TerraMind: Large-Scale Generative Multimodality for Earth Observation

Supplementary Material

In the following, we provide additional information on our data, the pretraining of TerraMind and its tokenizers, the quality of the tokenization, any-to-any generation matrices, and comparisons of TerraMind in unimodal and multimodal finetuning against specialized U-Net and ViT models.

7. TerraMesh Dataset

All versions of TerraMind have been pretrained on TerraMesh or a subset of it. TerraMesh is a comprehensive multimodal Earth observation dataset designed for large-scale model pre-training. It will be made publicly available under a permissive license in a preprint during the review process of this paper. The dataset includes nine modalities and we visualize examples of the dataset in Figure 8.

The dataset contains over 9 million globally distributed, spatiotemporally aligned samples across nine core modalities. Each modality is precisely co-registered at a 10-meter resolution, primarily based on Sentinel-2 grids. The S-1 and S-2 samples are sourced from MajorTOM-Core [23] and SSL4EO-S12 v1.1 [6]. It integrates Sentinel-1 SAR data with Sentinel-2 optical data (L1C top-of-atmosphere and L2A bottom-of-atmosphere reflectance), ensuring versatility for various downstream tasks. Because the source datasets contain only one S-1 product, each sample has either S-1 GRD or S-1 RTC data. Additionally, TerraMesh includes normalized difference vegetation index (NDVI) maps derived from Sentinel-2, Copernicus digital elevation model (DEM) data providing topographic context, and land-use/land-cover (LULC) maps from ESRI, enhanced with accurate cloud masks generated by the SEnSeI v2 model[22].

To ensure broad geographic and thematic diversity, TerraMesh employs subsampling techniques, selectively including representative samples from each global ecoregion and land-cover class, while downsampling highly homogeneous regions such as deserts and tundra. Another critical aspect is the data preprocessing pipeline, which includes reprojection, temporal alignment, and filtering to minimize missing data and artifacts, ensuring high-quality, analysis-ready samples.

TerraMind.v1-B-single was pre-trained on a subset of TerraMesh with one million samples, specifically the SSL4EOS12 v1.1 locations, using only four image modalities: S-2 L2A, S-1 GRD, DEM, and LULC. Additionally, we performed continuous pre-training with image captions. These captions were created using LLaVA-Next [37] and Overture Maps data [47]. The automated captioning pipeline includes a prompt with a chain-of-thought process to generate diverse captions. The captioning model is asked to generate three question-answer pairs and describe the full image later. We use the S-2 RGB bands and Overture base layer tags as inputs. Domain experts evaluated a subset of 1.3k captions, resulting in 69% of the captions without any hallucinations while the average completeness scores were 3.87 on a scale from 0 to 5.

8. Pretraining details

In this section, we give additional details on the pretraining of both TerraMind and its tokenizers.

8.1. Tokenizer models

The tokenizer models are pretrained using a Vision Transformer (ViT) encoder and a patched UNet decoder, with input images ranging from 224x224 to 256x256 in size. The model was trained with patch sizes of 16x16 for the ViT encoder and 4x4 for the UNet decoder. A tanh MLP was used before the quantizer, as outlined in the ViT-VQGAN paper, to enhance tokenization quality.

The model utilized a Finite-Scalar Quantization (FSQ) approach with a codebook size of 8-8-8-6-5, aiming to learn consistent and abstract representations across image patches. The latent dimension was set to 5. We leverage the normalization of codebook entries to the unit sphere during training. This concept is borrowed from the ViT-VQGAN approach, which applies a specific form of normalization to improve the quality and efficiency of learned representations. Additionally, an EMA-based quantizer was used with a decay rate of 0.99 to track and improve quantization over time.

During diffusion-based pretraining, the model was trained for 1000 timesteps using a linear beta schedule, with MSE loss as the objective. The training leveraged half-precision (fp16) and used an AdamW optimizer with specific learning rate scheduling and warmup strategies. The model also incorporated model EMA for stable training and set a batch size of 1 per GPU with various regularization techniques like grad clipping and random horizontal flips.

We pretrained the TerraMind tokenizers for image-like modalities with DDP on 4 GPUs for a total of 100 epochs on the respective modality of TerraMesh. We use a base learning rate of 1e-4, an effective batch size of 64 samples per GPU, i.e. the global batch size is 256. We reach a GPU utilization of 99% for single channel modalities like LULC and NDVI, and over 80% for all multi-channel modalities.

8.2. TerraMind

We pretrained both TerraMindv1-B and TerraMindv1-L with DDP on 32 GPUs. We determine the global batch size based on initial experimental runs comparing a global batch size of

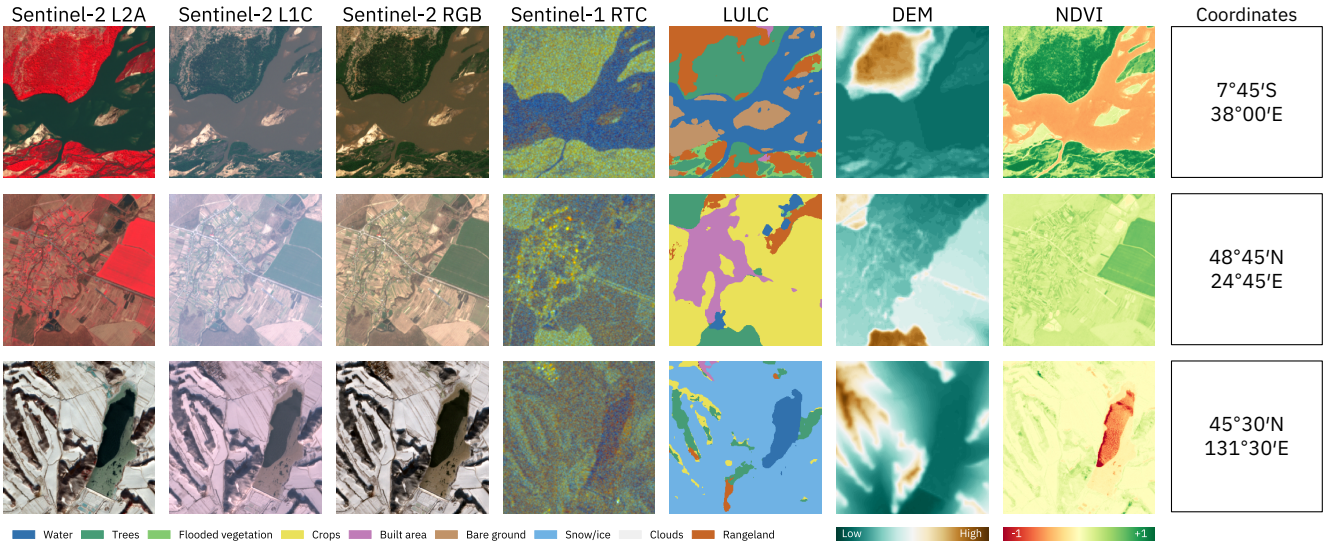


Figure 8. Visualization of the spatial-temporal alignment across modalities in TerraMesh. S-2 L2A uses IRRG pseudo-coloring and S-1 RTC is visualized in db scale as VH-VV-VV/VH. Copernicus DEM is scaled based on the image value range.

2K, 4K, and 8K. In addition, we determine the base learning rate starting from 1e-4 and iteratively experimented with half and double learning rates. Ultimately, we end up with a base learning rate of 2e-4 for a cosine annealing scheduler set to run for 500B tokens. For the v1-L model, we reach a GPU utilization of 85+%. Overall, the training of TerraMindv1-B took 12 days on 32 A100 GPUs, i.e., 9'216 GPU hours. Over the course of the pretraining, we also experiment with different configurations of the Dirichlet sampling distribution. In total, the pretraining experiments have been approximately three times larger than the final runs resulting in approximately 30K GPU hours allocated for pretraining.

We provide an overview on the scaling dynamics when going from TerraMindv1-B to TerraMind v1-L in Figure 9 with identical hyperparameters and compute. Overall, as expected, we observe a significant gap in the validation losses across modalities. We finally provide the validation losses per modality after pretraining of TerraMindv1-B and TerraMindv1-L in Table 9.

Model	S-2 L2A	S-1 GRD	S-1 RTC	DEM	NDVI
Random	9.68	9.68	9.68	9.68	9.68
V1-B	5.67	7.84	7.64	2.19	6.42
V1-L	**5.34**	**7.69**	**7.53**	**2.14**	**6.25**

Table 9. Validation losses of full pre-training of TerraMindv1-B and v1-L.

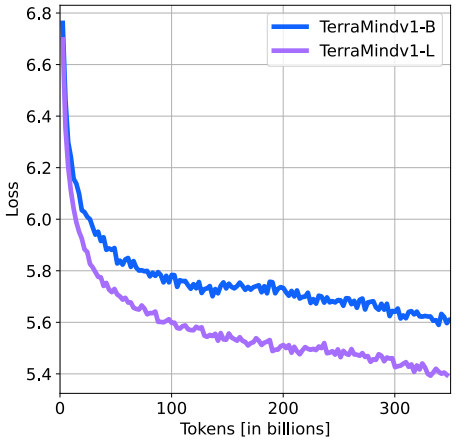


Figure 9. Example of the scaling behavior of TerraMind comparing v1-B and v1-L models for the first 350B tokens on the validation loss of optical S-2 L2A data. Overall, TerraMind-L outperforms TerraMind-B after approximately 10% of the training schedule of the large model.

9. Tokenizer performance and general learnings

In the following, we provide details on the tokenizations of TerraMind. At least for image-like modalities, the tokenizations represent an important and computationally heavy phase of the pretraining, which is why we highlight important learnings in the following.

Learnings. Overall, we learned that the tokenizer performance can be quite sensitive, which is especially related

to the significant bottleneck compression of up to 3000x after the encoder. When leveraging finite-scalar quantization (FSQ) instead of vector quantization (VQ), we observed exactly what the original FSQ paper [51] claims: FSQ makes quantization easier – yet in our experiments it did not improve the reconstruction performance in terms of MSE losses. We leverage FSQ as the training was more stable and less sensitive to the learning rate, which is likely related to the fact that, unlike VQ, FSQ does not require an additional codebook loss. We still observed that all tokenizer models were sensitive to the learning rate, with higher learning rates resulting in non-differentiability (NaN losses), and low learning rates caused blurry results.

In addition, we experimented with the codebook size. In our experiments, we observed that the level of detail in the reconstructions was significantly higher for single channel input compared to multi channel input (e.g., 12 band S2-L2A data). Naturally, with less channels, the compression bottleneck for equal-sized codebooks is lower. Therefore, we hypothesized whether multi-spectral data requires larger codebook sizes to obtain higher level of detail in the reconstructions. In contrast to our expectation, when increasing the codebook size over 16K for modalities with more than three input channels, the reconstructions had significant artefacts. This suggests that even though the compression bottleneck is lower, higher codebook sizes are more difficult for the model to use, which is in line with previous literature. However, we were surprised to see more artefacts in the reconstructions of models with a codebook size 32K compared to 16K.

Finally, we experimented with exponential moving average (EMA) updates for the tokenizer models. As expected, the models were less responsive to gradient updates. The resulting reconstructions smoothed out more of finegrained features. Together with the generative diffusion process in the tokenizer decoder, the resulting reconstructions often looked like hallucinations, e.g. bridges over rivers were not existing anymore in the reconstruction images. We therefore decided to ommit expotential moving average in our tokenizer models.

9.1. FSQ vs. VQ

Generally, our pretraining experiments comparing FSQ with vector quantization suggest that both approaches can achieve the same level of performance, yet reaching optimal levels of performance with VQ is regarded to be more challenging than using FSQ. We visualize this through (a) the reconstruction loss and (b) the gradient norms of the tokenizer pretraining on S-2 L2A data in Figures 10 and 11, respectively. Overall, we observe that both approaches reach the same level of convergence, however FSQ requires less tuning and is generally more stable than VQ. This especially also applies for the grad norms.

Performance. In the following, we assess the accuracy of our tokenizer models. Besides visual quality assessments and quantitative assessments with MSE metrics, we were particularly interested in whether our tokenizers exhibit geospatial biases. Understanding this is crucial to ensure TerraMind has a uniform level of performance across the globe. In addition, we investigate the reconstructions of radar data in more detail, as radar data by nature includes significant noise in the amplitude data. This could interfere with the noise generation in the diffusion process of the decoder, which is why we assess the structure of the reconstructions using SSIM and PSNR metrics.

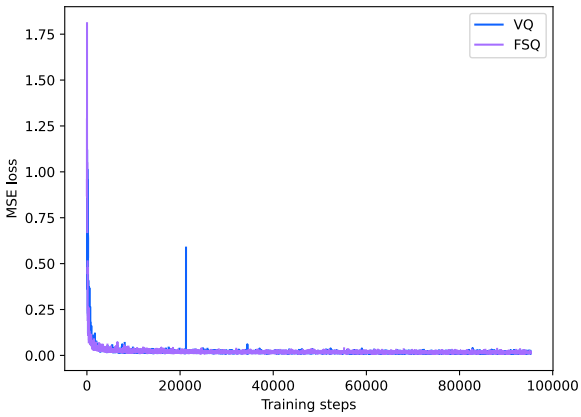


Figure 10. Pretraining reconstruction losses of S-2 L2A modality comparing finite-scalar quantization (FSQ) and vector quantization (VQ) approaches. Overall, both approaches reach the same level of performance. The FSQ approach converges smoother than VQ, while requiring less tuning.

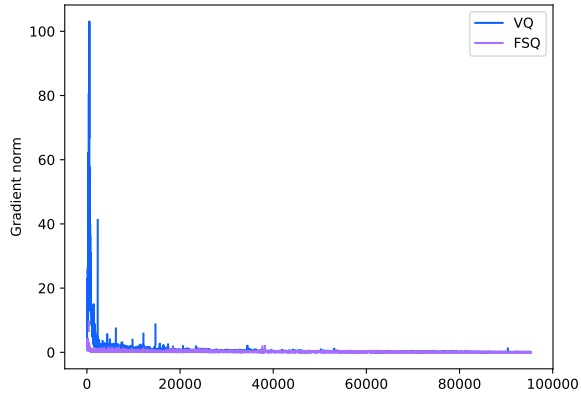


Figure 11. Gradient norms for pretraining of S-2 L2A tokenizers comparing finite-scalar quantization (FSQ) and vector quantization (VQ) approaches. The FSQ approach converges smoother than VQ, while requiring less tuning.

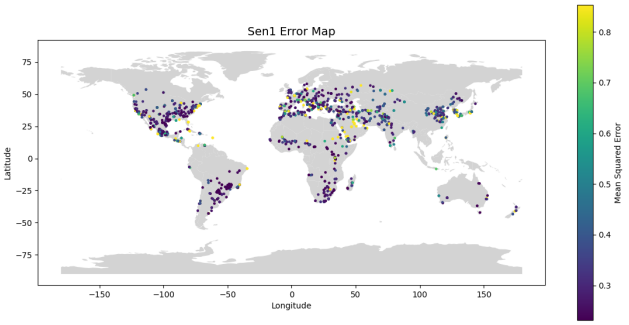


Figure 12. Spatial distribution of mean squared errors of the S-1 tokenizer on the validation set of the pretraining data.

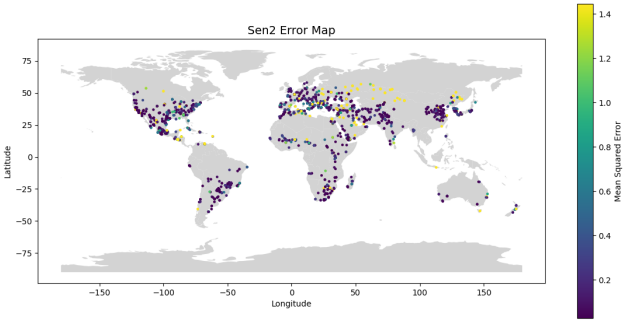


Figure 13. Spatial distribution of mean squared errors of the S-2 tokenizer on the validation set of the pretraining data.

In Figures 12 to 14, we provide an overview on the spatial distributions of the S-1 GRD, S-2 L2A, and DEM tokenizer on the validation data of the SSL4EO-S12 subset which is focused on urban areas and therefore relevant for many downstream applications. Overall, we observe low MSE errors and particularly low deviation across geographic regions. For optical S-2 data, we observe minor difficulties in reconstructing images from Northern Asia, which we manually investigated. Overall, the vast majority of those samples are depicting snowy/icy conditions that have very high reflectance values of up to 12,000 compared to a normal range of [0, 255] in RGB data. On those long tail distribution samples, the S-2 tokenizer naturally has more difficulties.

S1-tokenizer quantitative analyses. In the following, we pay particular attention to the performance of the radar S-1 tokenizer, which might be more challenging to train on a reconstruction task due to the inherent speckle noise in radar satellite data. We therefore evaluate the reconstructions of the S-1 tokenizer using the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR). Both input and reconstruction for S-1 are in a dB scale. In addition to S-1 evaluation metrics being computed in the dB space in Table 10, they also are calculated in the denormalized space. On the contrary, the S-2 evaluation metrics are computed in the normalized space.

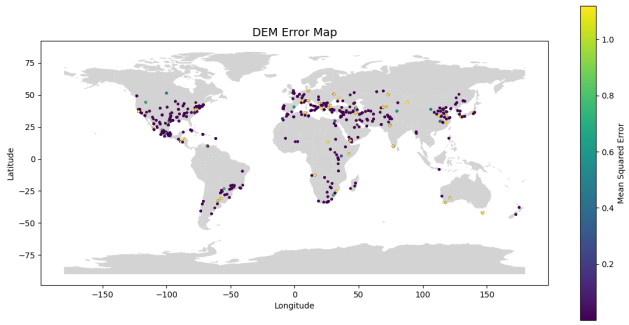


Figure 14. Spatial distribution of mean squared errors of the DEM tokenizer on the validation set of the pretraining data.

We give a more extensive background on radar data in the following for interested readers and non-EO experts. Reconstructing realistic and accurate synthetic aperture radar (SAR) S-1 VV and VH data is challenging due to factors inherent in the specific characteristics of SAR and the S-1 mission. SAR data is affected by complex interactions between the radar signal and Earth's surface. SAR is based on radar backscatter, which is influenced by surface roughness and moisture content. The interaction of radar waves with different surfaces, including vegetation structure and urban environments, can produce complex backscatter patterns. The two polarizations, VV and VH, capture different scattering mechanisms: VV is sensitive to surface roughness and vegetation, while VH captures cross-polarized interactions that are influenced by surface and volumetric features [14, 35, 56]. In addition, SAR inherently contains speckle noise, which obscures fine details, making it difficult to extract accurate information. To evaluate the SAR data tokenizers of TerraMind, we employ various evaluation metrics to assess quality and accuracy. We compute the MAE and RMSE for quantifying pixel-level differences, the SSIM to compare image structural content, and the PSNR [1, 67, 73].

Table 10 presents the quantitative evaluation of the TerraMind tokenizer reconstructions across multiple modalities. The results show a reasonable reconstruction performance for optical data, indicating both structural and perceptual fidelity. For radar modalities, S-1 GRD and S-1 RTC achieve comparable PSNR values, though SSIM scores are lower, suggesting that while the reconstructions are visually plausible, they exhibit moderate structural deviations. In addition to these quantitative metrics, we also conducted qualitative assessments through visual inspection to identify artifacts and inconsistencies not captured by numerical scores alone.

10. Additional experiments

In the following, we provide additional experiments, especially with regard to the quality of the latent space and the full finetuning performance. To understand the quality of the

Modality	MAE	RMSE	SSIM	PSNR
S-1 GRD	2.403	3.220	0.565	30.291
S-1 RTC	2.216	2.888	0.466	30.389
S-2 L2A	0.055	0.134	0.851	27.439
DEM	170.7	737.2	0.974	20.712
NDVI	0.091	0.168	0.647	21.517

Table 10. Evaluation of SAR VV and VH and S-2 reconstructions by the TerraMind tokenizers using MSE ↓, SSIM ↑ and PSNR ↑ on the validation dataset of the SSL4EO-S12 subset (8.5k samples).

latent space, we compute performances of nearest neighbor approaches for image classification tasks or using prototypical neural networks. We assess the performance of full fine-tuning by comparing with end-to-end trained, task-specific models like U-Nets and ViTs. We additionally compare the quality of the generations with the pseudo-labels used to pretrain TerraMind in an ablation experiment in a zero-shot setup.

10.1. Geolocation prediction

To better understand how TerraMind assigns geolocations, we further employ a Monte-Carlo sampling on the latitude-longitude grid for an optical tile from the validation data in Figure 15. We observe that while TerraMind is not predicting the correct geolocation (●), there is a very high likelihood that the predicted geolocation is one of the adjacent grid points that have been seen during pretraining (●). This result suggests that even for data from unseen geolocations, TerraMind remembers similar samples from the pretraining data (●) and returns the geolocation of the samples with high similarity. This capability paired with the global pretraining of TerraMind suggests that geo-localization of data from unseen locations is possible but determined by the similarity to images from adjacent locations.

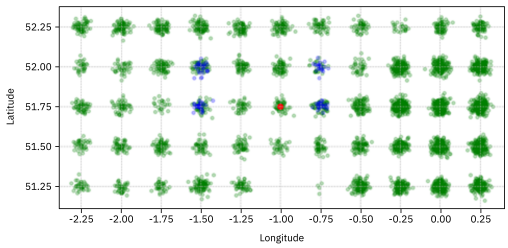


Figure 15. Distribution of predicted geo-locations for an optical S-2 L2A sample from the validation set. ● is the correct location, ● are Monte-Carlo sampled locations from TerraMind, ● represents the distribution of training locations. TerraMind's geo-localization seems to be based on similar optical samples in the training dataset for which TerraMind then outputs the geolocation.

We further extend the analysis of Figure 7 by additionally prompting the model for likely locations of urban areas. Overall, we observe that the model correctly identifies many densly populated areas across the globe. We also note over-predictions in, for example, North Africa and middle-east. This observation suggests that the model might confuse bare land and urban areas in these regions.

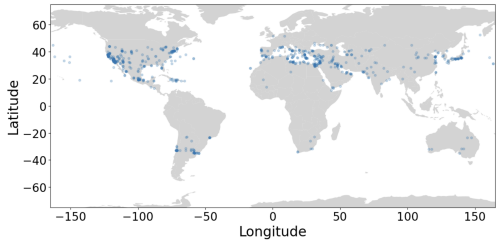


Figure 16. Prediction distribution of the land use class "urban" with a sampling temperature of $T = 1.0$. TerraMind has a reasonable internal representation of the geolocation of specific contexts, like land use classes.

10.2. Few-shot experiments

We present additional few-shot experiments with the EuroSAT and METER-ML dataset in Table 11. We use the embeddings of the pre-trained encoders without any additional fine-tuning. The patch embeddings of each image are averaged for image-level classification tasks.

The experiments include four different few-shot settings with varying numbers of examples and classes. 5-way refers to sampling five classes per run, while full-way describes experiments with all dataset classes per run. 1-shot and 5-shot indicate that one or five images are sampled for each class per run. 5-shot experiments with five support samples per class are using Prototypical Networks [60] for classification. This approach averages the embeddings of the selected labeled images (support set) and classifies the target images (query set) based on the class prototype with the lowest Euclidean distance from each sample. In the 1-shot setting, Prototypical Networks are mathematically equal to 1-Nearest-Neighbor classification. We refer to the original paper for details [60]. Different from literature, we evaluate each run on the full test set instead of subsampling query images.

TerraMind performs best on both datasets, outperforming all other geospatial foundation models as well as the CLIP vision encoder [57]. Interestingly, the base version leads to overall better results than the large model. Similarly, Prithvi's smaller 1.0 version has comparable results to its larger 2.0 300M version, indicating that model size has only a limited effect on few-shot performance.

In addition to S-2 L1C, the METER-ML dataset provides high resolution RGB images from NAIP with 1 m resolution. Only CLIP and TerraMind can process RGB images without any fine-tuning. While CLIP profits largely from the higher resolution inputs, TerraMind only performs marginally better

Model	Input	EuroSAT 5-way 1-shot	EuroSAT 5-way 5-shot	EuroSAT full-way 1-shot	EuroSAT full-way 5-shot	METER-ML 5-way 1-shot	METER-ML 5-way 5-shot	METER-ML full-way 1-shot	METER-ML full-way 5-shot
CLIP-ViT-B/16	S-2 RGB	57.00	70.72	43.92	58.30	29.15	37.44	23.13	30.53
CLIP-ViT-B/16	NAIP	–	–	–	–	32.01	42.35	25.66	35.81
DeCUR	S-2 L1C	50.54	64.35	37.53	50.82	27.87	33.64	20.95	27.21
Prithvi 1.0 100M	S-2 L1C	60.11	73.29	46.86	60.66	26.08	35.81	22.33	29.21
Prithvi 2.0 300M	S-2 L1C	61.06	73.21	47.47	60.47	28.26	36.13	22.52	29.59
TerraMindv1-B	S-2 L1C	**70.83**	**87.94**	**57.48**	**79.66**	**33.90**	43.89	**26.85**	<u>37.41</u>
TerraMindv1-B	NAIP	–	–	–	–	32.23	<u>44.75</u>	25.53	37.85
TerraMindv1-L	S-2 L1C	<u>70.07</u>	<u>86.29</u>	<u>56.58</u>	<u>77.39</u>	<u>33.09</u>	42.72	<u>26.02</u>	36.34
TerraMindv1-L	NAIP	–	–	–	–	32.59	**44.99**	25.94	**38.29**

Table 11. Few-shot classification results on EuroSAT and METER-ML measured in mean accuracy ↑ averaged over 200 runs. 5-way refers to five randomly sampled classes per run, which is a default setting used in few-shot learning. Full-way refers to sampling all dataset classes, i.e., ten EuroSAT classes and seven METER-ML classes. We highlight the best two models in bold and underlined.

and sometimes worse than with multispectral S-2 data. Notice that TerraMind shows similar performance gaps as CLIP when comparing NAIP data to S-2 RGB. This indicates that additional multispectral channels have a comparable effect on few-shot performance as high-resolution images.

10.3. Finetuning comparisons with baseline models

Since the first approaches to foundation models for Earth observations, experts in the field discuss on the usability of such models compared to task-specific models that are trained for each application individually. Recent benchmark results suggested that task-specific models, like U-Nets, often outperform finetuned GFMs [49]. We therefore additionally investigate how TerraMind compares with task-specific U-Nets and ViT models following the PANGAEA evaluation protocol in Table 6. As adviced by the authors of PANGAEA, we again report results on nine of the eleven datasets as we could not reproduce the performance on the remaining two datasets. The task-specific models are trained from scratch for each individual task, while all GFMs including TerraMind are finetuned with a frozen encoder and an UperNet head. Overall, our results demonstrate that TerraMindv1-B outperforms task-specific UNet and ViT models across the PANGAEA benchmark in both unimodal and multimodal settings by 1pp avg. mIoU and 4pp avg. mIoU respectively. In multimodal settings, the improvement peaks to 4.5pp improvement of TerraMindv1-B over task-specific U-Nets. To the best of our knowledge, this is the first time a GFM model outperforms task-specific models on a global benchmark.

In addition, we observe that for most datasets, TerraMindv1-B outperforms TerraMindv1-B-single. This demonstrates the benefit from scaling in the data and feature dimension–i.e., leveraging dual-scale feature representations on a pixel level and a token level.

10.4. Comparing generations and pseudo-labels

We evaluate the model generations for modalities where we used pseudo-labels as input data. For example, in initial experiments with TerraMindv1-B-single, we leverage Google's DynamicWorld model to pseudo-label LULC maps which we use as input to the model. In the following experiment in Table 12, we test the performance of the DynamicWorld model against the generations of TerraMind. Overall, we observe that while finetuned TerraMindv1-B-single outperforms DynamicWorld, the generation of TerraMind does not surpass the inference results of DynamicWorld.

Approach	**Input**	**IoU**$_{Water}$
TerraMindv1-B-single	S-2 L1C	69.87
Dynamic World pseudo-labeling	S-2 L1C	71.98
TerraMindv1-B-single finetuning	S-2 L1C	**76.32**

Table 12. Results on the Sen1Floods11 test set comparing flood maps derived from TerraMind's out-of-the-box LULC generations to those derived from LULC pseudo-labeling with Dynamic World. The results are inferior to those obtained by fine-tuning a specialized model for this downstream task, which is expected.

10.5. TiM tuning for crop mapping

We further investigate the relevance of TiM tuning for crop type mapping in order to understand the relevance of generating artificial data for more finegrained segmentation tasks. That means, we generate artificial LULC data which includes agricultural crop as a *single class* and investigate whether this additional information helps to segment nine different types of crops in satellite images. We experiment with the South Africa Crop Type Mapping dataset (https://source.coop/esa/fusion-competition) and present the results in Table 13. Overall, we observe that

TiM tuning improves the performance by around 1pp. That means that even though the generated artificial data does not include further information on the location and shape of certain crops, the information on where to expect crop land in general helps to guide the model to an improved performance.

	Input	mIoU
TerraMindv1-B	S-2	41.87
TerraMindv1-B TiM	S-2 + *gen. LULC*	**42.74**

Table 13. Thinking-in-modalities (TiM) tuning compared with standard full fine-tuning approaches on the SA Crop dataset.

11. Any-to-any generation

In Figure 18, we provide an example of any-to-any generation on four image-like modalities and two sequence-like modalities. Overall, we observe that when we start from modalities with high information content (e.g., fine-grained image-like modalities), the reconstructions are particularly good. Even with less information content, the model is able to generate consistent artificial data. However, we can clearly observe that the quality compared to the ground truth (represented by the input in the left of the figure) is decreasing. Finally, it is interesting to see how artefacts are introduced by the model when starting from lower information content in the input. For example, when promting TerraMind to generate data from DEM input, we observe that the model pays significant attention to the darker streams in the DEM image, which are later generated as a river in LULC.

While we expect to see accurate generations from information-rich modalities like optical data, it is particularly interesting to understand how TerraMind deals with low information content. Therefore, we prompt TerraMind to generate a subset of modalities starting from the geolocation in Figure 17. Interestingly, for a geolocation from the middle-east, the model generates an optical image that resembles a desert. While the generated optical image is based on the right context, the actual structure is unsurprisingly different from the ground truth. Based on the chained generation, this difference ripples down across all other modalities as well causing consistent but inaccurate generations. This example emphasizes the relevance of access to information-rich, fine-grained features to facilitate accurate generations.

Next to the evaluation of raw, pixel-level input in Table 3, we further evaluate the generation quality using tokenized input in Table 14. Interestingly, we observe only minor reduction in performance compared to pixel-level input even though the tokenized representations are compressed significantly (up to 3000x for S-2 L2A). Overall, our results suggest that leveraging tokenized inputs can be a reasonable alternative to leveraging pixel-level data for the generation of artificial data with TerraMind.

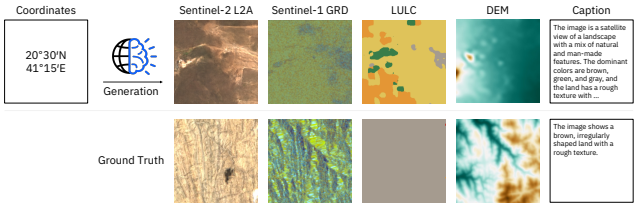


Figure 17. Randomly selected chained generation example with uni-modal geo-location input data. Top row is artificially generated data by TerraMind, buttom row represents a ground truth sample at this grid location, respectively.

11.1. Large-scale generations

In Figures **??** and **??**, we provide additional qualitative results for large-tile generations at the example of Singapore. Specifically, we leverage a 35.5km $\times$ 69.5km optical S-2 L2A tile as input and iteratively generate overlapping 224x224 pixel generations for S-1 RTC, S-1 GRD, NDVI, and LULC. In the overlapping areas, we apply the mean of all generations in order to enhance the spatial conciseness of the generations. TerraMind consistently removes the clouds in the S-1 generations. It makes assumptions for hidden areas, which are look accurate for large features like water bodies or the shore line. Other features like airports or ships are also clearly visible in the S-1 and NDVI generations.

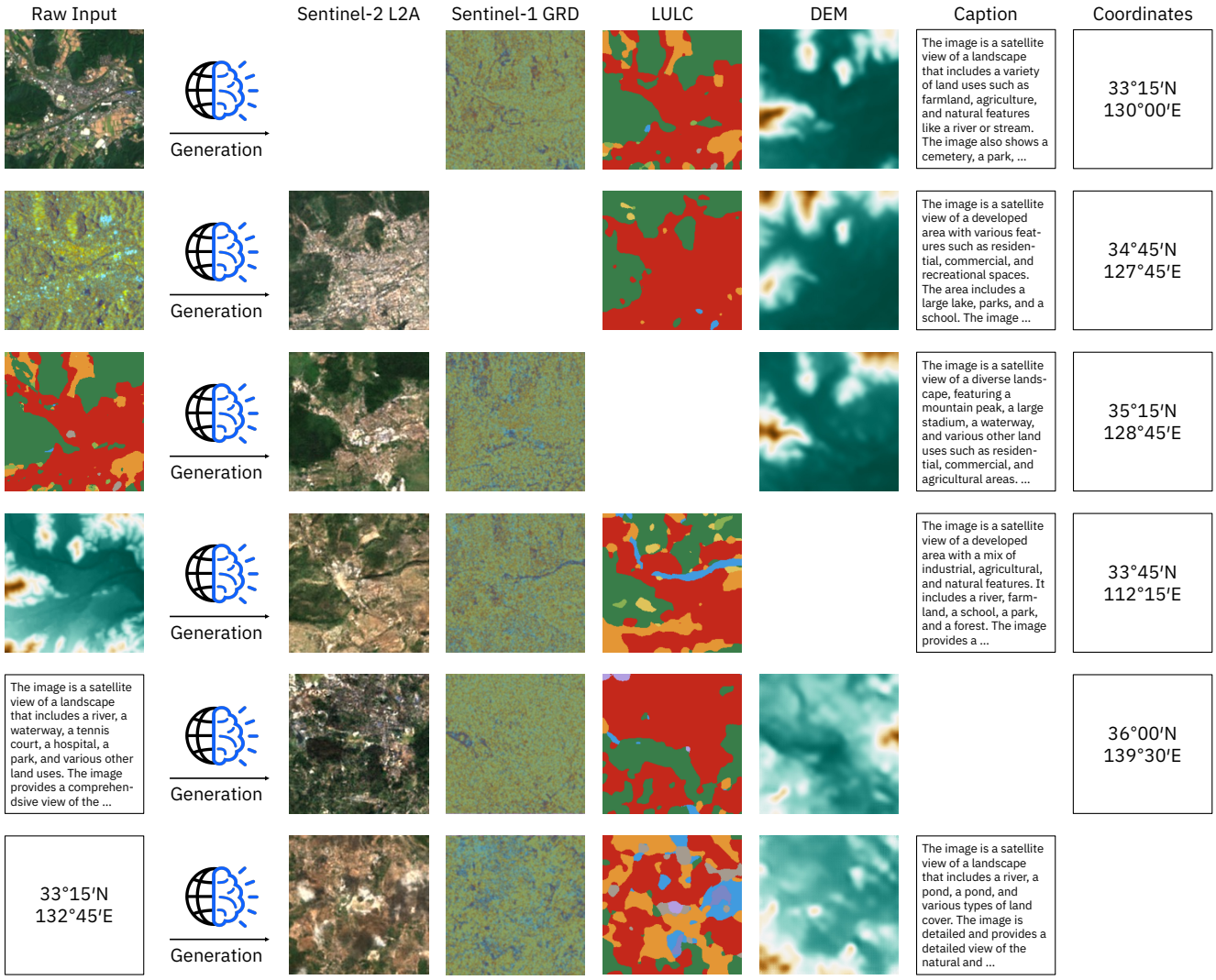
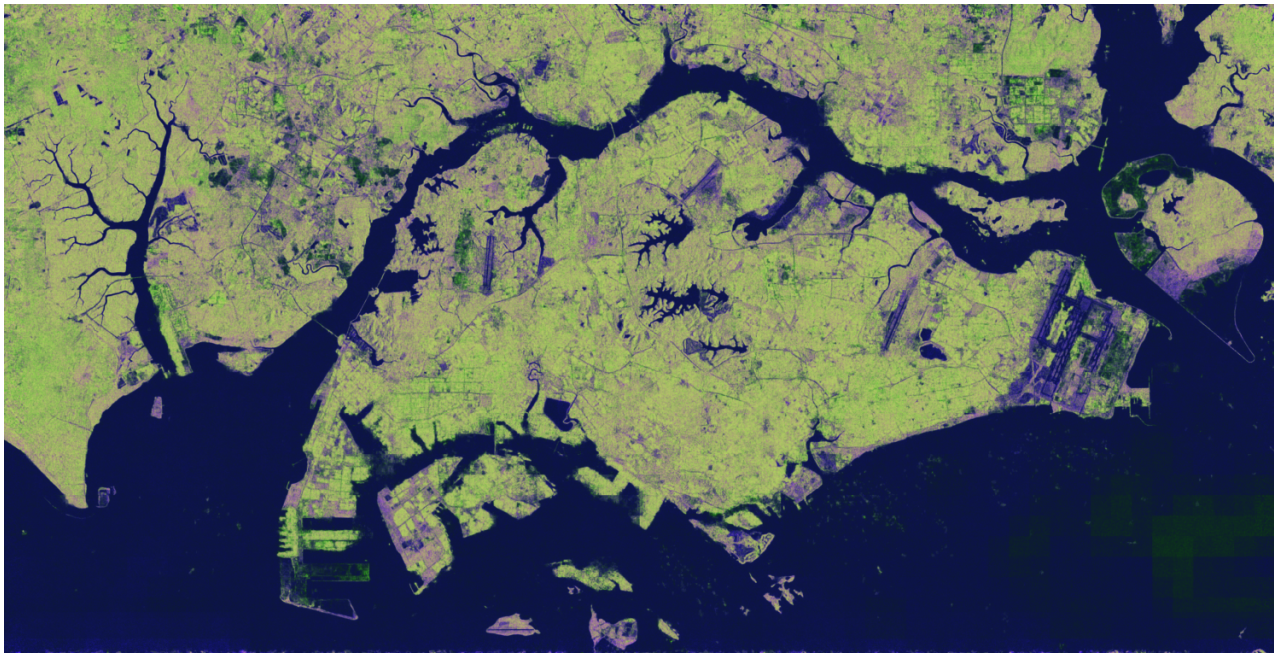


Figure 18. Any-to-any generation example of TerraMindv1-B-single. Fine-grained input like optical and radar achieve particularly good performances.

Modalities	**MAE**	**RMSE**	**SSIM**	**PSNR**
Tokenized S-2 L2A → S-1 GRD	3.3180	4.3309	0.5131	27.715
Tokenized S-2 L2A → S-1 RTC	3.0544	3.9178	0.4131	27.739
Tokenized S-2 L2A → DEM	572.5	1040.6	0.5728	17.718
Tokenized S-1 GRD → S-2 L2A	0.0820	0.1238	0.7182	25.630
Tokenized S-1 GRD → NDVI	0.1949	0.2425	0.4124	18.324
Tokenized S-1 GRD → DEM	327.4	550.3	0.7271	16.008
Tokenized S-1 RTC → S-2 L2A	0.1195	0.1935	0.6638	24.266
Tokenized S-1 RTC → NDVI	0.1895	0.2348	0.4500	18.606
Tokenized S-1 RTC → DEM	457.9	851.6	0.7095	19.457

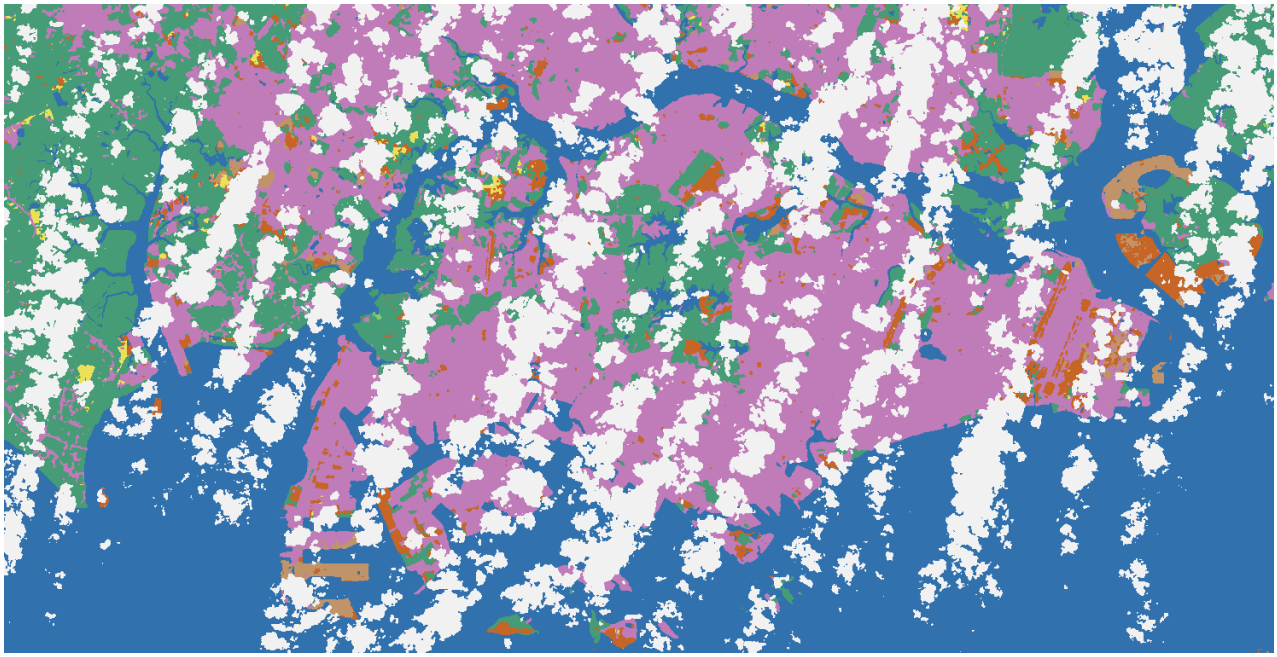
Table 14. Performance of TerraMind on tokenized inputs using 10 diffusion steps. Metrics include MAE ↓, RMSE ↓, PSNR ↑, and SSIM ↑.

(a) Input: S-2 L2A data from Singapore captured January 9th, 2025.



(b) Generation: TerraMind output for S-1 composition

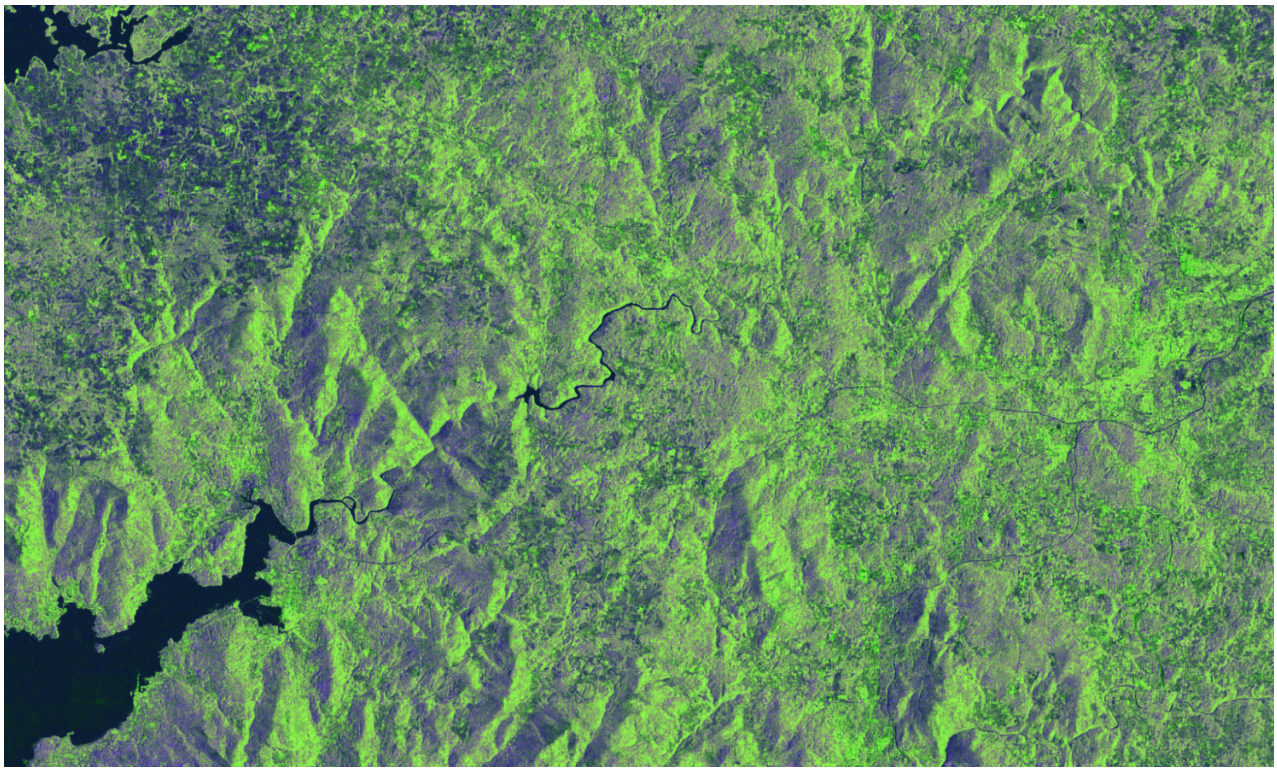
Figure 19. Large-tile generations of TerraMind for Singapore (1/1)



(c) Generation: TerraMind output for LULC

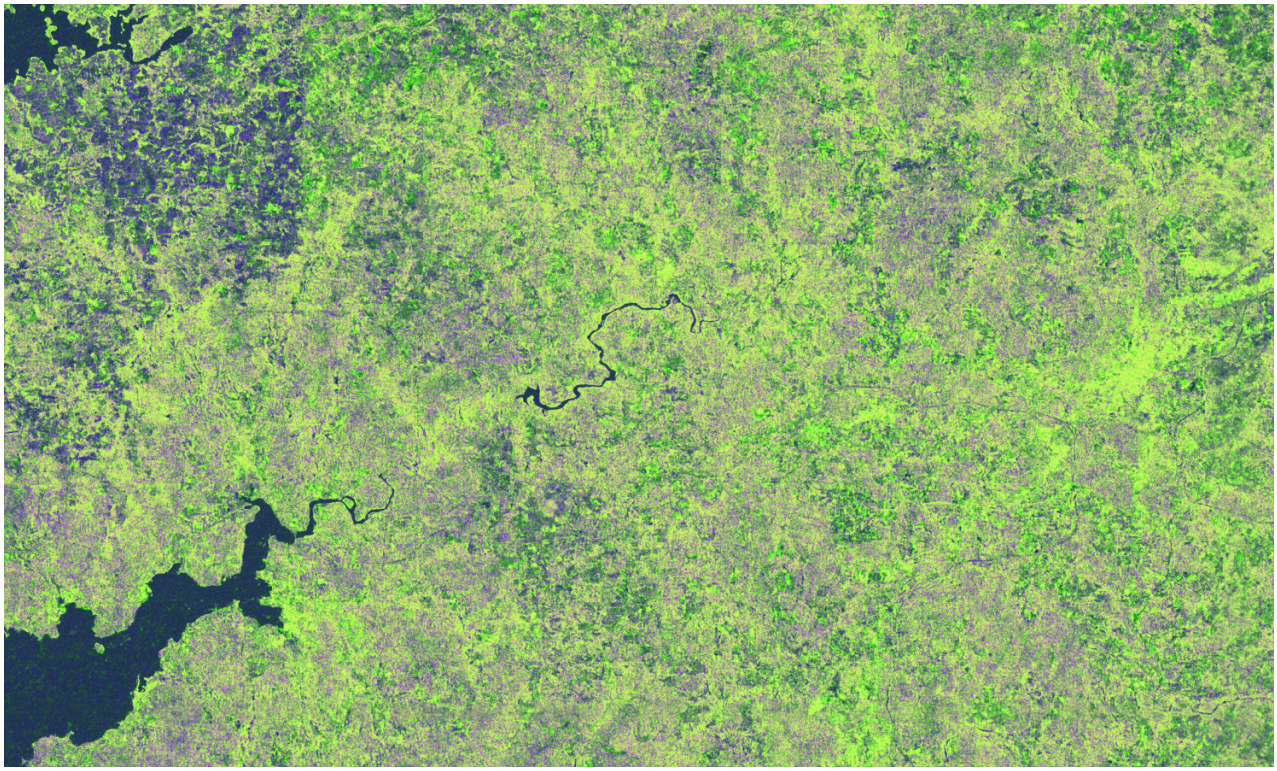
Figure 19. Large-tile generations of TerraMind for Singapore (2/2)

(a) Input: S-2 L2A data from Santiago de Compostela.

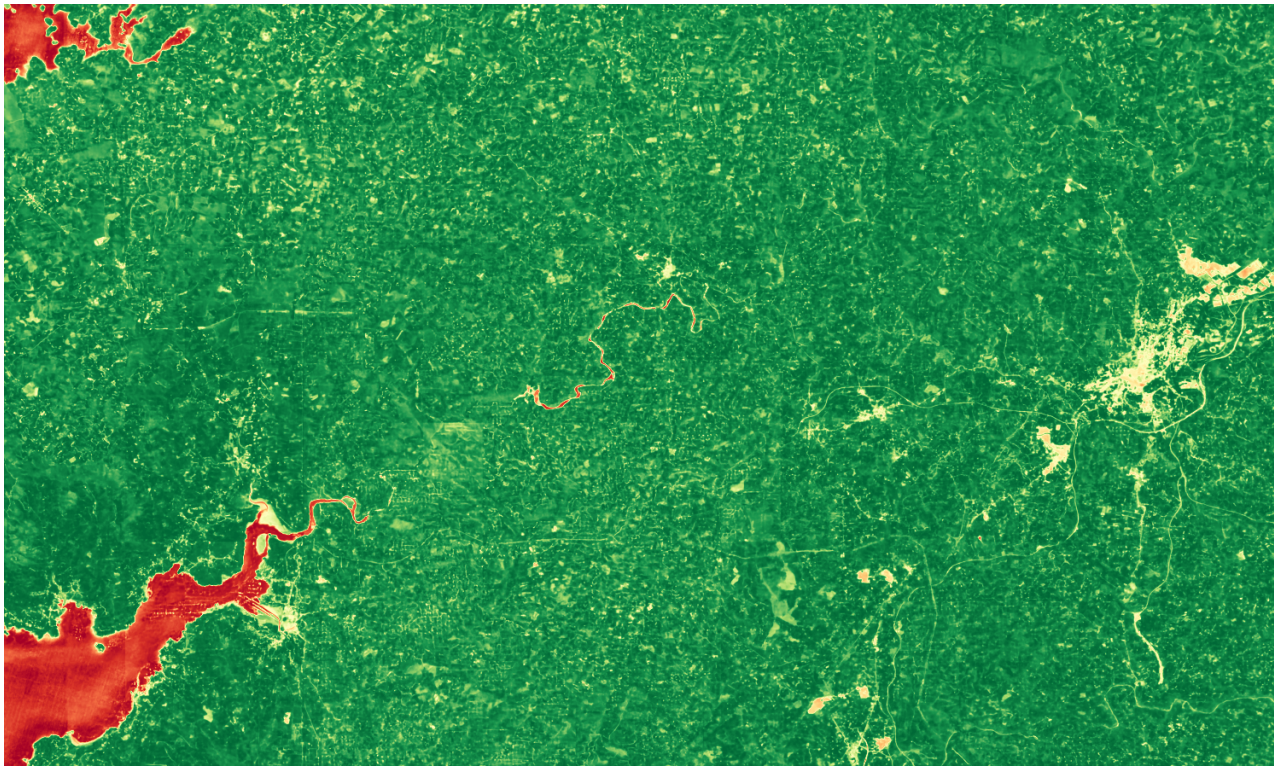


(b) Generation: TerraMind output for S-1 GRD composition

Figure 20. Large-tile generations of TerraMind for Santiago de Compostela (1/3)

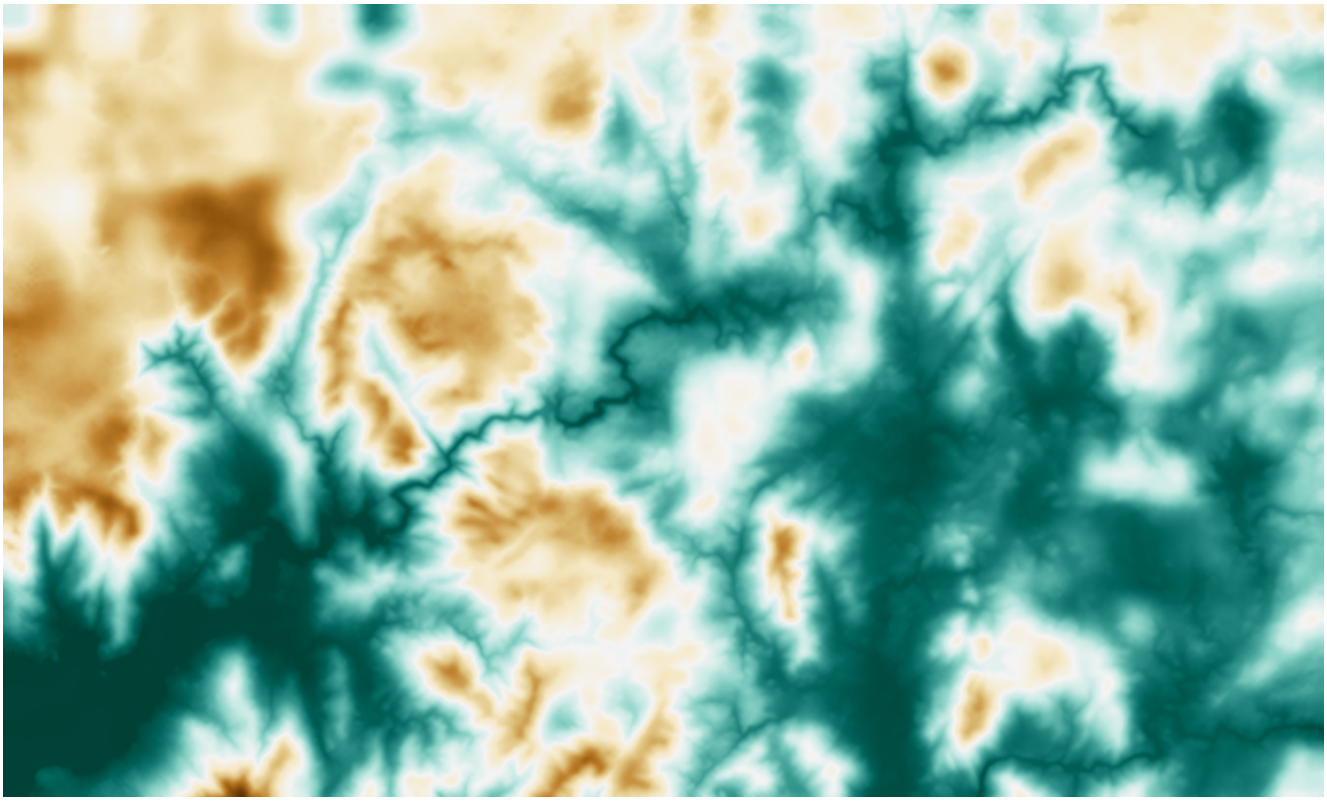


(c) TerraMind generation for S-1 RTC composition



(d) Generation: TerraMind output for vegetation

Figure 20. Large-tile generations of TerraMind for Santiago de Compostela (2/3)



(e) Generation: TerraMind output for digital elevation

Figure 20. Large-tile generations of TerraMind for Santiago de Compostela (3/3)