

PYViT-FUSE: A FOUNDATION MODEL FOR MULTI-SENSOR EARTH OBSERVATION DATA

Manuel Weber*, Carly Beneke

EarthDaily Analytics
1055 Canada Pl #33,
Vancouver, BC V6C 3L5, Canada
`{manuel.weber, carly.beneke}@earthdaily.com`

ABSTRACT

We propose PyViT-FUSE, a foundation model for earth observation data explicitly designed to handle multi-modal imagery by learning to fuse an arbitrary number of mixed-resolution input bands into a single representation through an attention mechanism. The learned patch tokens are further processed by a stack of vision transformers with a novel pyramidal structure. We train the model on a globally sampled dataset in a self-supervised manner, leveraging core concepts of the SwAV algorithm. We show the interpretability of the fusion mechanism by visualization of the attention scores and the models applicability to downstream tasks.

1 INTRODUCTION

Foundation models (FM) for earth observations (EO) have gained traction following the success of large language models (LLM) and their demonstration of scaling laws (Kaplan et al., 2020). The premise is that training larger models on vast datasets enhances performance. This idea has been central to computer vision, where datasets like ImageNet (Deng et al., 2009) have enabled pre-training in both supervised and unsupervised settings, leading to breakthroughs in model design and training. While pre-trained encoders extract valuable features from both natural and satellite images, key differences exist: Natural images use three bands (RGB), whereas satellite sensors capture multiple wavelengths, often generating images with tens or even hundreds of bands (hyperspectral data). Additionally, satellite image resolution varies significantly across sensors, complicating or even preventing direct model transfer.

To address these challenges, we introduce PyViT-FUSE, a model designed to process multi-source, multi-modal images at their native resolutions. It enables flexible band selection through a channel fusion approach leveraging attention mechanisms. Unlike many EO foundation models that employ masking for self-supervised learning (Reed et al., 2023; Astruc et al., 2024; Jakubik et al., 2023), we adapt SwAV (Swapping Assignments between Views) (Caron et al., 2021), eliminating the need for a decoder and pixel-space reconstruction. This avoids a key limitation of MAEs (Masked Autoencoders) (He et al., 2021), which rely on accurate pixel-space reconstructions—a challenge in EO due to the less constrained nature of satellite imagery. As shown in Section 2, SwAV also enables embeddings independent of input band combinations, allowing the same model to be applied across diverse downstream tasks. For instance, Sentinel-2 data may be sufficient for object detection but limited by cloud occlusion; integrating Sentinel-1 data mitigates this issue using the same model. We explore this important capability for a concrete segmentation task in appendix B.2.

2 METHODOLOGY

In this section we describe our proposed model architecture (section 2.1) and explain how we adapt the SwAV algorithm as self-supervised learning approach (section 2.2). A core concept we leverage in this work is that multi-modal EO data is available co-registered for a given location, covering

*Corresponding author (`manuel.wbr@gmail.com`)

the same surface patch. We will refer to such a patch as area of view (AOV) with size $H \times W$ (in meters) which is the same for all considered modalities. Note that by modality we mean the data from a specific satellite source which can contain multiple bands, often at different resolutions.

2.1 MODEL ARCHITECTURE

The architecture of our model can roughly be divided into three main components: 1. Input module, 2. Fusion module, and 3. Pyramidal vision transformer as illustrated in Fig. 1. Each of these components plays an important role in achieving the overall goal of generating embeddings enriched with extracted information from the available modalities.

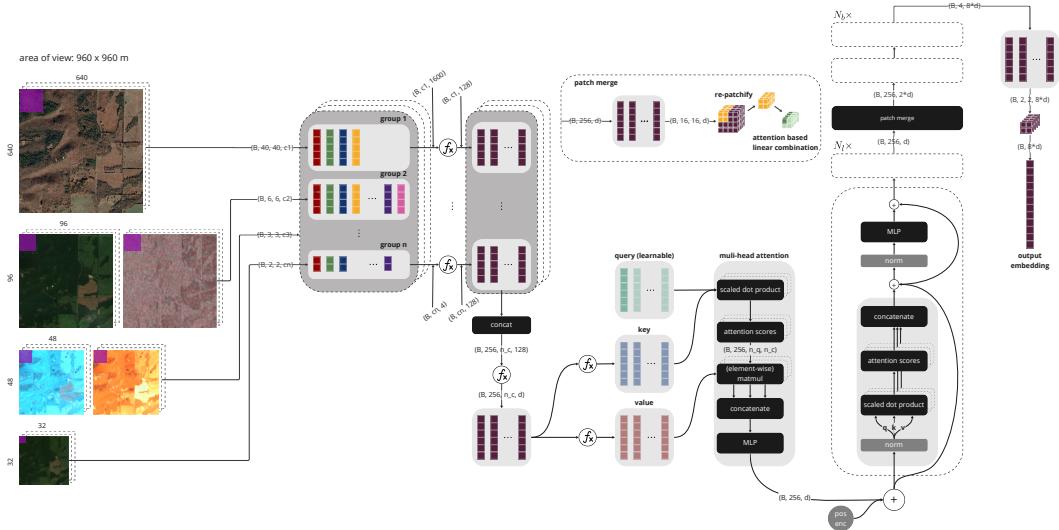


Figure 1: Model architecture of PyViT-FUSE consisting of three main components. An embedding representing the multi-modal input is generated independent of band combination.

Input Module. The input module processes each modality by dividing images into patches and transforming them into a shared feature space. Let X_i^r be a 2D input array of size $H^r \times W^r = H/r \times W/r$ for band i at resolution r . It is split into $n_p \times n_p$ patches, flattened into vectors of size $H^r/n_p \times W^r/n_p$, denoted as x_i^r . Grouping all x_i^r from modality M_j forms the subset $S^{M_j} = \{x_0^{r_0}, x_1^{r_0}, x_2^{r_1}, \dots, x_n^{r_m}\}$, while the complete input set is $S = \{S^{M_0}, S^{M_1}, \dots, S^{M_k}\}$, where k is the number of modalities, each with n_j bands, for a total of $n_B = \sum_{j=0}^k n_j$. A single modality may contain bands at different resolutions r_0, r_1, \dots, r_m . Processing all bands at their native resolution without resampling is a key feature of our approach. Each raw pixel vector x_i^r is then transformed into a common feature space via two learnable linear projections \mathcal{P}_i^1 and \mathcal{P}_i^2 , yielding $\hat{x}_i = \mathcal{P}_i^2(\mathcal{P}_i^1(x_i^r))$ with feature dimension d . Using two projections instead of one allows setting a hidden layer dimension d_r such that both layers have comparable parameter counts, ensuring sufficient capacity to learn transformations independently of band resolution.

Fusion Module. The input module outputs a batch tensor of shape (B, n_p^2, n_B, d) , where B is the batch size. To feed this into the vision transformer, which processes a sequence of patch tokens, the band dimension is fused into a single representation via attention. The fused representation is a weighted sum of input bands, $\hat{x} = \sum_i w_i \hat{x}_i$, where weights w_i correspond to attention scores. A multi-head cross-attention layer performs this fusion: key and value vectors are derived from linear projections of input tokens, while the query is a learned vector. The query size is arbitrary but should be large enough to effectively learn band fusion. Each attention head uses the same query but learns individual attention scores (w_i). The outputs from all heads are concatenated and passed through a linear layer. As shown in Section 3, this method enables visualization of band importance at each patch position, as attention weights determine how much information from each band propagates into subsequent model layers.

Transformer Module. At this stage, the data is a tensor of shape (B, n_p^2, d) , processed through multiple multi-head self-attention layers. We follow the transformer encoder of Vaswani et al. (2023) but use pre-norm instead of post-norm, as in Xiong et al. (2020). Additionally, we introduce a pyramidal structure where, after N_l self-attention layers, spatial patch merging is applied, similar to CNN pooling (Krizhevsky et al., 2012). This uses the fusion module to merge patches via attention, increasing feature dimension by a factor of s . As illustrated in Fig. 1, patches are first reassembled spatially, then re-patchified into $s \times s$ groups before fusion. Like band fusion, this results in a linear combination of neighboring feature vectors weighted by attention scores. Setting $s = 2$ reduces the sequence length by a factor of four after each merge step. Repeating N_l self-attention layers followed by patch merging N_b times yields an output feature map with $n_p/2^{N_b-1}$ feature vectors of dimension $d \cdot 2^{N_b-1}$. This pyramidal structure compresses information into abstract representations, enabling downstream tasks that utilize feature maps at different depths.

2.2 MODEL TRAINING

We adopt SwAV as a self-supervised training method, where the encoder generates similar embeddings for augmented views of the same input. Unlike contrastive methods (Chen et al., 2020), SwAV aligns embeddings to prototype vectors instead of relying on negative samples. Fig. 2 illustrates SwAV, and we refer to Caron et al. (2021) and appendix A.2 for details. While the original paper uses multi-crop augmentation to generate local views, this is unsuitable for satellite images, where small crops may have distinct semantics. Instead, we exploit remote sensings unique multi-sensor observations, generating local views by randomly dropping modality sub-sets $S_i^{M_j}$ and individual bands x_i^r . Dropped bands are replaced with a learnable empty token, as shown in Fig. 2. This augmentation strategy enables SwAV to handle multi-sensor data, leading to a model that generalizes across different input band combinations. We carefully tune hyper-parameters such as the softmax temperature, learning rate, number of prototype vectors (set to 512 in this work) and batch size to avoid mode collapse.

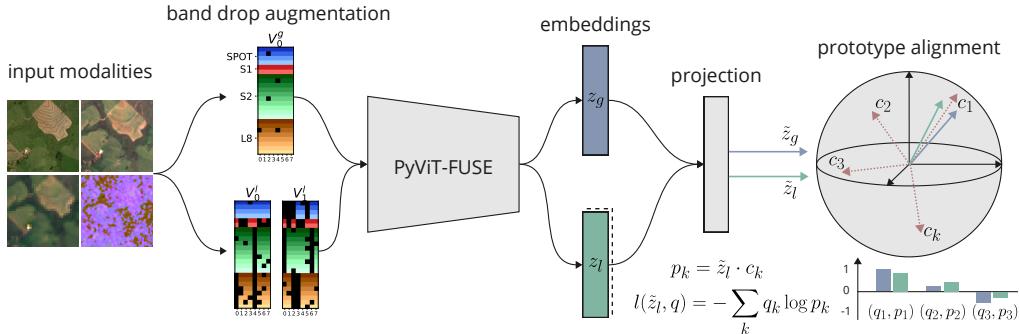


Figure 2: Illustration of band drop data augmentation and self-supervised training with SwAV algorithm. Band drop is visualized by coding the sensor as color hue and its bands as different shades for 8 batches. Blacked out parts indicate dropped channels.

Dataset and training details. For a proof of concept, we curated a globally sampled dataset of $\sim 1M$ samples (see appendix A.1 for a map of the sample distribution). We choose an AOV of 960 m and include the following sensors/modalities: *Airbus SPOT* (4 bands), *Sentinel-1* (2 bands), *Sentinel-2* (10 bands) and *Landsat-8* (8 bands) for a total of 24 bands with resolutions ranging from 1.5 to 30 m/pixel. The models internal representation dimension is set to $d = 128$, the query size of the fusion modules to 4096, and it consists of $N_l = 8$ transformer layers per block and $N_b = 4$ total blocks. All attention layers (self- and cross-attention) consist of 8 heads. This configuration results in a model with $\sim 103M$ parameters (3.8M for the input module, 1M for the fusion module and 98M for the ViT). We train the model on 8 A10G GPUs with a batch size of 256 using a stochastic gradient descent (SGD) optimizer with momentum of 0.9 for 30 epochs. The total training time was ~ 1 week or ~ 1344 GPU hours.

3 EXPERIMENTS AND RESULTS

We first inspect the feature maps generated by the trained model at the output of each ViT pyramid block. Fig. 3 shows an input sample where we display each modality as RGB image and the corresponding feature maps averaged across the feature dimension (see appendix B.1 for more samples and detailed feature maps). It is clear that the model has learned to extract spatial features well.

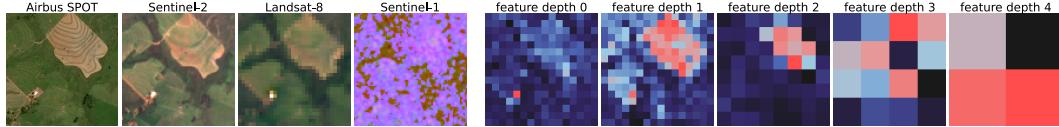


Figure 3: Sample input visualized as RGB image for each modality and corresponding averaged feature maps at the output of each ViT pyramid block. Colors indicate low (black) to high (red) activations.

A more quantitative assessment of the models learned skills is to inspect the regularity of the output embeddings. The objective of the learning task is for the model to generate embeddings that are close together in feature space if the inputs contain similar information as is the case for different band combinations of the same input sample. Given a global view of sample i (V_i^g) and a local view of sample j (V_j^l) within a batch, we can determine the cosine similarity $\sigma(V_i^g, V_j^l)$ as well as L^2 distance $d_2(V_i^g, V_j^l)$, where we expect σ to be high and d_2 to be low if $i = j$ and opposite if $i \neq j$. Fig. 4 illustrates both metrics as a matrix where i, j correspond to the rows and columns. The images demonstrate that the model has learned to generate similar embeddings for views of the same surface patch with different band combinations which is the expected behavior.

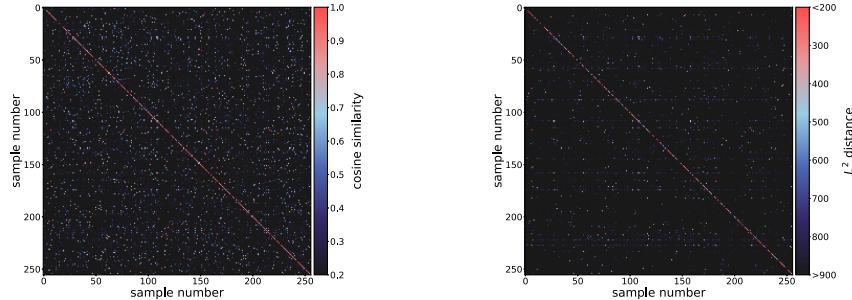


Figure 4: Cosine similarity (left) and L^2 distance (right) between the embeddings of the global and a local view for a sample against all other samples in the batch.

The fusion model allows the visualization and interpretation of the relative importance of the various bands at a given location. Fig. 5 show the attention scores of each head of the fusion module for the same sample as in Fig. 3. We use a color coding that represents the sensor and band which has the highest attention score at the respective location, hence indicating the importance of the respective band. Visually, it is apparent that different heads learned to pay more attention to certain bands for specific features of the input data which is consistent across samples (see appendix B.1).

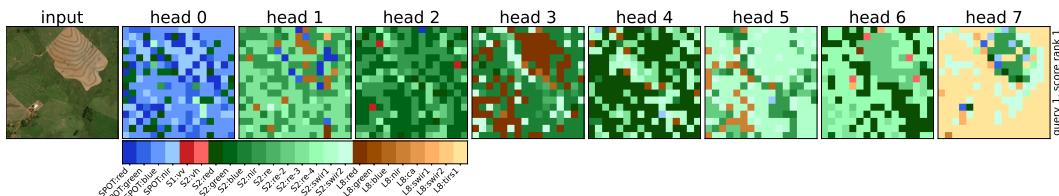


Figure 5: Visualization of attention scores for each head of the fusion module. The color corresponds to the band with the highest score indicating the corresponding importance of the band.

4 CONCLUSION AND OUTLOOK

We demonstrate a novel approach for the fusion of mixed-resolution, multi-modal satellite imagery, its interpretability due to attention mechanism and leverage an innovative concept for self-supervised training of a large model on vast amounts of data. As the next step, we will run a number of benchmark tests to quantify the model capability across different tasks as well as scale the methodology.

REFERENCES

- Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Omnisat: Self-supervised modality fusion for earth observation, 2024. URL <https://arxiv.org/abs/2404.08351>.
- Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2021. URL <https://arxiv.org/abs/2006.09882>.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL <https://arxiv.org/abs/2002.05709>.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- K.S. Fujita, Z.H. Ancona, L.A. Kramer, M. Straka, T.E. Gautreau, C.P. Garrity, D. Robson, J.E. Diffendorfer, and B. Hoen. United states large-scale solar photovoltaic database (v2.0, august, 2024): U.s. geological survey and lawrence berkeley national laboratory data release, 2023. URL <https://energy.usgs.gov/uspvdb/>.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL <https://arxiv.org/abs/2111.06377>.
- Johannes Jakubik, Sujit Roy, C. E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, Daiki Kimura, Naomi Simumba, Linsong Chu, S. Karthik Mukkavilli, Devyani Lambhate, Kamal Das, Ranjini Bangalore, Dario Oliveira, Michal Muszynski, Kumar Ankur, Muthukumaran Ramasubramanian, Iksha Gurung, Sam Khallaghi, Hanxi, Li, Michael Cecil, Maryam Ahmadi, Fatemeh Kordi, Hamed Alemohammad, Manil Maskey, Raghu Ganti, Kommy Weldemariam, and Rahul Ramaachandran. Foundation models for generalist geospatial artificial intelligence, 2023. URL <https://arxiv.org/abs/2310.18660>.
- 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, 2020. URL <https://arxiv.org/abs/2001.08361>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL <https://arxiv.org/abs/1412.6980>.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning, 2023. URL <https://arxiv.org/abs/2212.14532>.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL <https://arxiv.org/abs/1706.03762>.

Ruiben Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. URL <https://arxiv.org/abs/2002.04745>.

A MODEL AND TRAINING DETAILS

A.1 DATASET

For a proof of concept, we curated a dataset that includes four modalities (Airbus-SPOT, Sentinel-1, Sentinel-2 and Landsat-8) with a total of 24 bands. We gathered $\sim 1\text{M}$ samples (a total of 5.8 TB) at random locations around the global where all data sources are available. We only select samples where the cloud fraction is $< 7\%$ for all modalities. We remove samples in arid regions at a rate of 0.8 since they are largely cloud free which would lead to an oversampling of very similar image patches. Fig. 6 shows the distribution of samples.

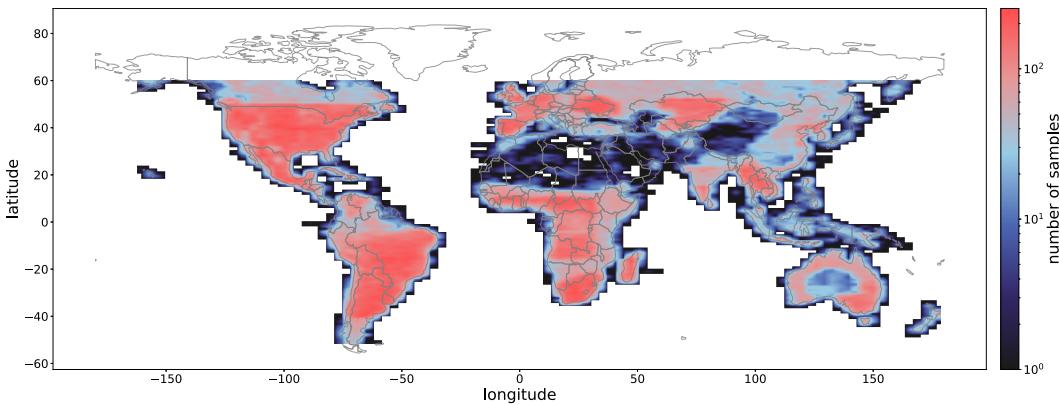


Figure 6: Distribution of samples gathered globally for an initial dataset.

A.2 MODEL TRAINING

Model training is performed using the self-supervised SwAV algorithm. We use a band drop augmentation and other image augmentations, such as additive and multiplicative noise, for generating global and local views. Global views should mostly contain uncorrupted inputs or weak augmentations, such as few single bands dropped, while the local views are created with heavy augmentations including dropping of entire input modalities. Training with SwAV requires n_g global views and $n_l > n_g$ local views from which the loss is calculated as

$$L = \sum_{i=0}^{n_g} \sum_{j=0}^{n_l} (1 - \delta_{ij}) l(\tilde{z}_j, q_i) \quad (1)$$

There are two mode collapses in which the model learns a trivial solution: 1. Always predict the same cluster assignment or 2. assign clusters uniformly across all prototype vectors, independent of the input data. The first case is addressed by using the Sinkhorn-Knopp algorithm to re-distribute the distribution of q . This method works best if the number of prototype vectors is $\ll B$ (batch size). This requirement invokes some compute limitations as the number of prototype vectors should be relatively high (thousands) which also requires the batch size to be high. This limitation can be somewhat reduced by introducing a queue that stores the projected embeddings from previous batches. Current prototype assignments are then re-computed for the current training step using the queue while the queue is updated with the current batch at the end of the step. In this work we choose a batch size of 256 and a queue size of 4 allowing the number of prototype vectors to be 512.

After the model is trained, we can verify its ability to assign the same prototypes to both the global and local view. Fig. 7 provides a visualization of the prototype assignments for three different samples. The top part of each strip corresponds to the codes provided by the global view while the bottom part are the corresponding predictions based on the local views. The strength of alignment against each prototype is indicated by the color of the line. This demonstrates that different samples are aligned with different prototypes, only a few prototypes have strong alignments and p and q are aligned with the same prototypes which is the expected behavior.

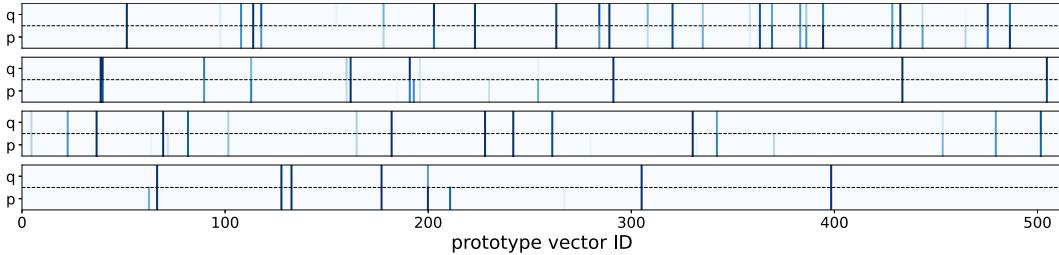


Figure 7: Visualization of alignment with prototype vectors for the global (q) and local (p) view of three samples.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 FEATURE MAPS AND ATTENTION SCORES

Fig. 3 shows the averaged feature maps at the output of each ViT block. This, however, does not illustrate all the learned features for a given output. In Fig. 8 we show all 128 feature maps at the output of the first ViT block illustrating the various spatial structures the model picks up on. Deeper feature maps are less interpretable since spatial components are further compressed.

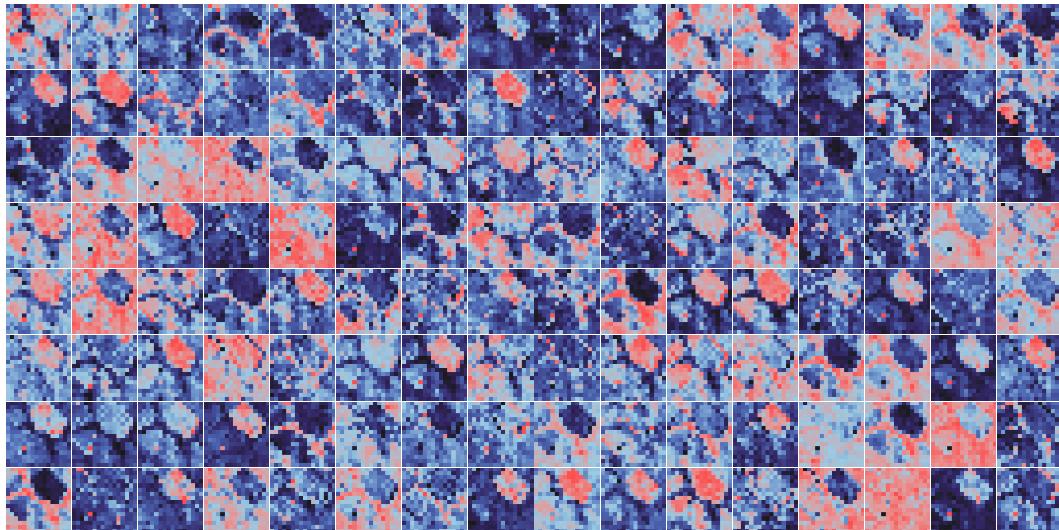


Figure 8: Map of all features from the output of the first ViT block, color coded from low (black) to high (red) activation.

We show additional input samples and corresponding averaged feature maps in Fig. 9, similar to Fig. 3. For the same samples we illustrate the band importance given by the highest attention scores in each head of the fusion module in Fig. 10. It is apparent that different heads tend to focus on specific modalities and bands as well as spatial features.

B.2 APPLICATION

We demonstrate a concrete application of our foundation model where we use a dataset containing the segmentation masks of large solar photovoltaic installations (Fujita et al., 2023). In total, the dataset contains 4185 facilities aggregated into 1597 samples across the United States, including 142 samples with no positive pixels, for which we gather the imagery from all modalities after but close to the facilities installation date. We randomly select 100 of the samples and set it aside for validation. We attach a simple decoder network which processes the encoder feature maps from all depths by a series of convolutional + up-sample + concatenate layers until the desired output

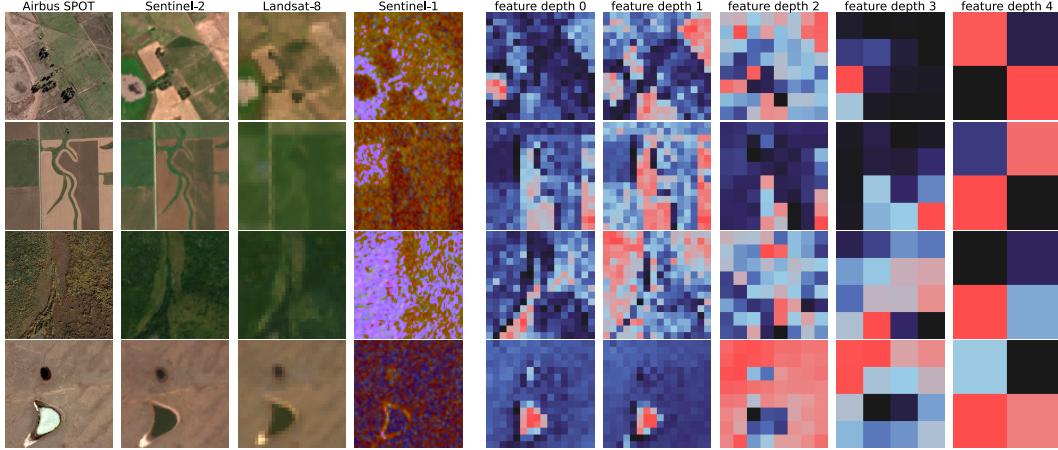


Figure 9: Visualization of the input as RGB image of four samples and corresponding averaged feature maps at the output of each ViT block.

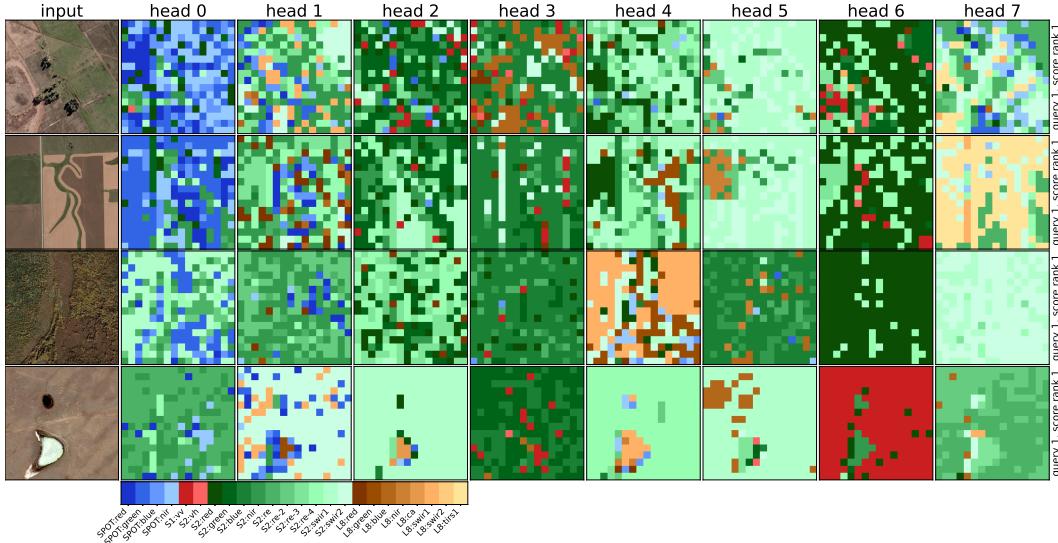


Figure 10: Visualization of the attention scores from all heads of the fusion module for four samples. The color represents the sensor and band with the highest attention score at the given position.

resolution is reached. This is commonly referred to as Feature Pyramid Network (FPN). As output resolution, we choose 10 m/pixel resulting in an output map of size 96×96 pixels. We train the network by minimizing the binary-crossentropy loss using the Adam optimizer (Kingma & Ba, 2017) in two steps: First, all weights are frozen except the decoder, resulting in 3.7M trainable parameters. After 50 epochs, we un-freeze the weights in the fusion module in order to adapt for this specific task. The second stage consists of 4.7M trainable parameters and continues for another 100 epochs. During training, the learning rate is decreased from an initial value of $1e^{-4}$ to $1e^{-7}$ according to a cosine function.

We perform the same training procedure for different input band combinations in order to study the effect of variable information content for solving the specific task. The first fine-tuned model uses all modalities as input while the second model only uses 6 bands (red, green, blue, NIR, SWIR1, SWIR2) of Sentinel-2 and 2 bands (VV, VH) of Sentinel-1. The third model further reduces the amount of input information and only uses 3 bands (red, green, blue) of Sentinel-2. We also train a CNN based FPN with a ResNet-50 encoder as backbone pre-trained on ImageNet as baseline. Since the pre-trained model expects three input bands, we use red, green and blue of

Sentinel-2. Fig. 11 illustrates the output probability maps of all models for four specific samples. They include various sizes of targets. The first two samples are considered easy as the input is very clean while the last two samples are hard due to cloud obstruction in the Sentinel-2 imagery. We also show the contour lines (yellow) of the binary output map at a threshold of 0.5. There are two main observations:

1. The detection accuracy increases as more bands are added to the input. This is quantified as the foreground IoU at an output threshold of 0.5 which increases from 0.33 for S2 (RGB) to 0.58 for S1, S2 (8 bands) and 0.68 for all sensors. The baseline model achieves an IoU of 0.44. This is further illustrated in Fig. 13 where we plot the training curves for the metrics accuracy, background IoU and foreground IoU during training of the various models.
2. For the inputs which contain obstructions in Sentinel-2 imagery, the model accuracy is generally worse. Both the baseline model and our model, using only Sentinel-2 RGB as input, perform poorly on these hard samples. However, the models which utilize bands from other modalities perform much better, highlighting the importance and benefits of fusing different modalities.

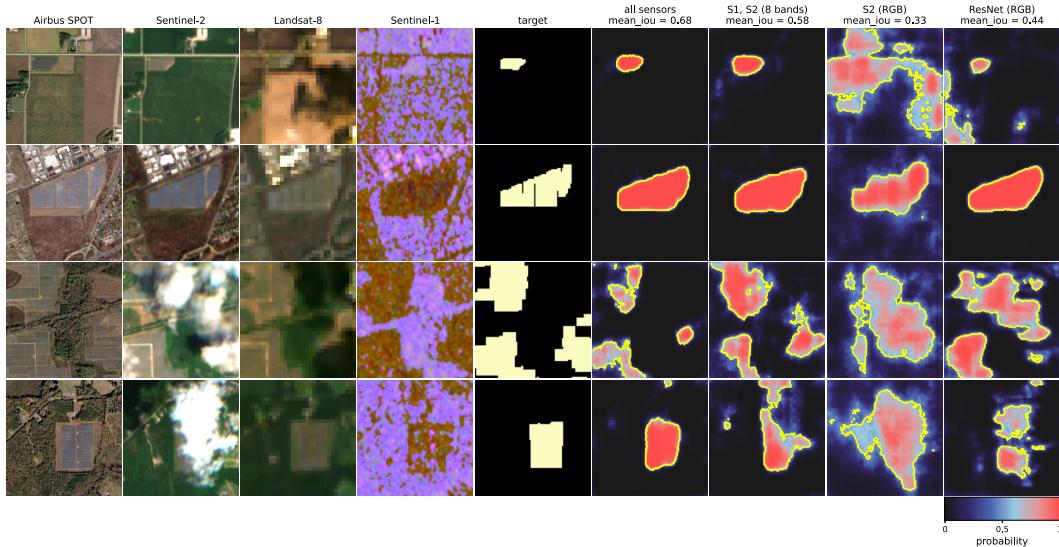


Figure 11: Input of all modalities visualized as RGB image (first four columns from the left), corresponding target map and the output probability maps of the fine-tuned PyViT-FUSE model on the solar PV dataset for three different input band combinations. The right column corresponds to the output probability map of a fine-tuned FPN with a pre-trained ResNet-50 encoder as a comparison.

The visualization of the attention scores of the fusion module is particularly interesting for the model using Sentinel-1 and Sentinel-2 as shown in Fig 12. It shows that the bands with highest attention scores belong to the modalities used as input while the empty tokens of all other bands do not carry any valuable information. Moreover, it shows that specific heads pick up on the locations of obstruction by assigning higher attention scores to the Sentinel-1 bands, notable in head 6 for the last two samples where the bands colored in red align with the clouds.

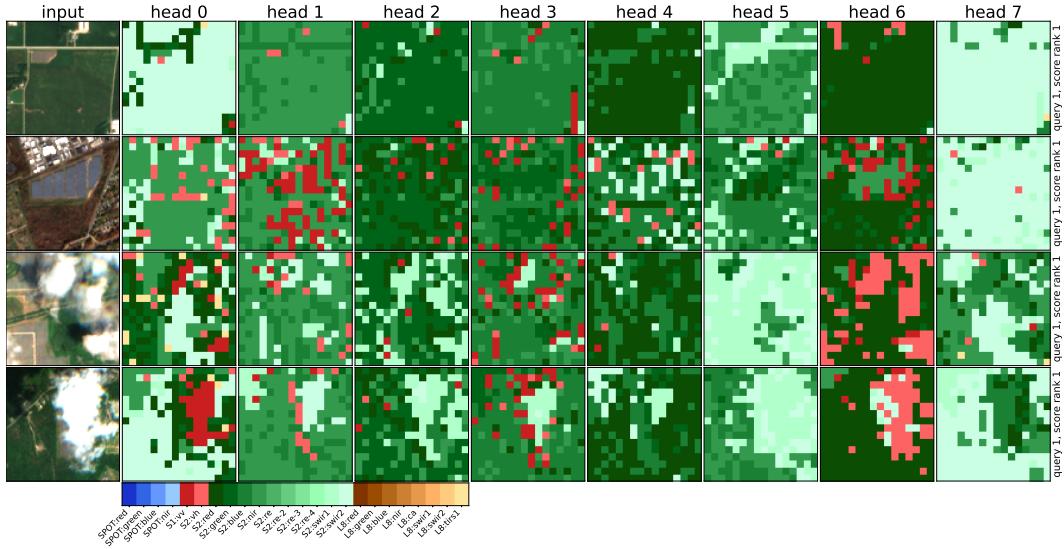


Figure 12: Visualization of the attention scores of the fusion module where the color corresponds to the band with the highest score. The input consists of 6 bands from Sentinel-2 and 2 bands from Sentinel-1.

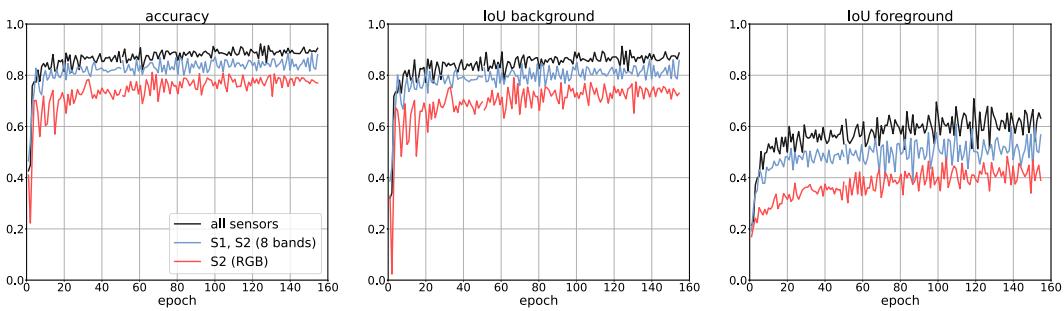


Figure 13: Training plots of accuracy, background IoU and foreground IoU from fine-tuning PyViT-FUSE on the solar PV dataset for three different input band combinations.