Sparse Multimodal Vision Transformer for Weakly Supervised Semantic Segmentation

Joëlle Hanna Michael Mommert Damian Borth
AIML Lab, School of Computer Science, University of St.Gallen
{firstname}.{lastname}@unisg.ch

Abstract

Vision Transformers have proven their versatility and utility for complex computer vision tasks, such as land cover segmentation in remote sensing applications. While performing on par or even outperforming other methods like Convolutional Neural Networks (CNNs), Transformers tend to require even larger datasets with fine-grained annotations (e.g., pixel-level labels for land cover segmentation). To overcome this limitation, we propose a weakly-supervised vision Transformer that leverages image-level labels to learn a semantic segmentation task to reduce the human annotation load. We achieve this by slightly modifying the architecture of the vision Transformer through the use of gating units in each attention head to enforce sparsity during training and thereby retaining only the most meaningful heads. This allows us to directly infer pixel-level labels from image-level labels by post-processing the unpruned attention heads of the model and refining our predictions by iteratively training a segmentation model with high fidelity. Training and evaluation on the DFC2020 dataset show that our method[1] *not only generates high-quality segmentation masks using image-level labels, but also performs on par with fully-supervised training relying on pixel-level labels. Finally, our results show that our method is able to perform weakly-supervised semantic segmentation even on small-scale datasets.*

1. Introduction

Over the past few decades, access to public data from Earth observing satellites has drastically improved, making it easier to automate large-scale land cover mapping through advances in machine learning and high-performance computing. However, this task remains challenging due to the scarcity of labeled data. The increasing complexity of state-of-the-art predictive models and the heterogeneity of land cover classes further compound this requirement. Traditional machine learning algorithms that rely solely on accurate labeled data (strong supervision) therefore have limited performance in this domain.

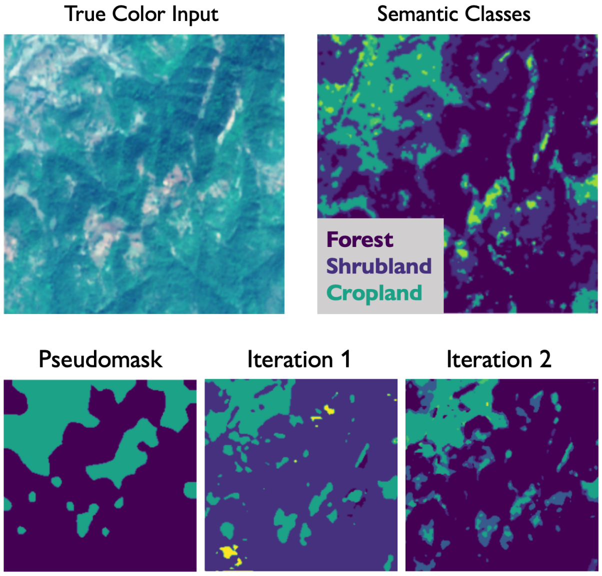

Figure 1. Our method uses training images with image-level labels (top left) to learn semantic segmentation (top right) in a weakly-supervised way. The second line shows the pseudomask obtained from the vision Transformer's sparse attention heads, as well as two iterations of refinement. We find that our segmentation method produces a result very similar to the ground truth.

Weakly supervised semantic segmentation (WSSS) aims to decrease the high cost associated with annotating ”strong” pixel-level masks by relying on ”weak” labels, such as bounding boxes [25], scribbles [12] and image-level labels [5]. This work aims to perform WSSS based on the least expensive, but most challenging label to use, which is the image-level class label. In this case, the standard WSSS process consists of three steps: first, training a multi-label classification model based on the image-level class labels; second, creating a binary mask for each class (pseudomask)

[1] Code is available on github.com/HSG-AIML/sparse-vit-wsss

and third, using this pseudomask to train a segmentation model in a standard, fully-supervised manner.

In contrast to previous approaches (see Section 2), we propose to explore representations learned by vision Transformer (ViT) models [6]. By examining the attention maps generated by ViTs, we can gain deeper insights into how the model actually perceives visual data. ViTs consist of multiple Transformer blocks, each containing multiple heads that project the input into different embedding subspaces to process various image features. Therefore, analyzing the attention maps of individual heads can help in understanding what particular regions of the image are being attended to by each head.

At the same time, several works have revealed that not all heads in a Transformer are equally important [17, 29]. As a result, one could potentially remove or ”prune” the inefficient heads and retain only the ones that have the most significant impact on the task. Studies conducted on NLP-domain Transformers [29] have shown that the use of a pruning technique relying on stochastic gates reduces the risk of removing important and meaningful heads. Consequently, this approach allows for the removal of most heads while maintaining the same level of performance.

This work aims to explore the use of weak supervision, specifically using image-level labels, as a solution to the challenge of replacing precise and therefore expensive groundtruth segmentation maps in land cover mapping (Figure 1). To achieve this, we propose to train a sparse vision Transformer for multi-label classification using learnable binary gating functions for the heads, which determine whether the head should be pruned or not. After training on image-level labels, we extract the representations from the remaining heads, cluster them, and construct pseudomasks serving as labels for training a segmentation model.

The contributions of this work are as follows:

- We demonstrate that in remote sensing, the vast majority of heads in vision Transformers can be removed without seriously affecting performance.
- We show that the remaining heads are meaningful and specialized, and can be utilized to infer pseudomasks for land cover mapping.
- In a comparison with the fully-supervised segmentation setup, we show that our weakly-supervised approach yields similar performances while only relying on image-level labels.
- Our work further illustrates that weak supervision combined with attention sparsity can effectively reduce the need of fine-grained labeled data, even on small-scale datasets.

2. Related Work

2.1. Weak Supervision.

Weak supervision is a branch of machine learning where training data is *imperfect*, as opposed to the traditional approach of fully-supervised learning where each example is expected to be annotated with precise and consistent labels. The supervision signals can come from various sources and may not always fully capture the underlying semantics of the image. In this section we will cover related works dealing with imprecise annotations including the case of coarse labels for learning semantic segmentation of images.

Recent interest in weakly-supervised semantic segmentation methods can be attributed to their simplicity and availability of noisy labels. Different types of annotations can be used for supervision, including bounding boxes [11, 25], scribbles [12, 27] or image-level labels [5, 30]. Usual steps consist of first training a model with the weak labels, generating a pseudo segmentation mask with potential refinement methods and finally training a segmentation model in a standard fully-supervised fashion. The most common method is to train a classification network using image-level labels as it requires minimal costs. Class activation maps (CAMs) [32] are typically used to generate pseudo labels for segmentation networks. However, some limitations to the activation maps acquired from the classification network include the fact that they only locate the most discriminative part of objects. To overcome this issue, Ahn and Kwak [2] propose to propagate CAMs' local responses to nearby areas that belong to the same semantic entity, using a semantic affinity network. Moreover, Wang *et al.* [30] use view consistency to encourage the semantic consistency between CAMs obtained from different spatial perturbations of the same image. Meanwhile, Ahlswede *et al.* [1] investigate the use of weak supervision techniques for tree species classification using remote sensing images, by exploring different methods such as CAM or GradCAM.

Our approach differs from previous work in that it leverages the representation learned by the vision Transformer directly, with minimal reliance on post-processing techniques.

2.2. Model Sparsification

The process of sparsification involves removing useless connections or elements within a neural network in order to decrease its size. Through pruning, not only can the number of parameters in trained networks be significantly reduced, but in certain cases, the overall accuracy can also be improved.

There are two categories of pruning methods: unstructured and structured. Unstructured pruning involves removing unimportant weights in the network by setting them to 0, based on predefined criteria like their magnitude [9, 10].

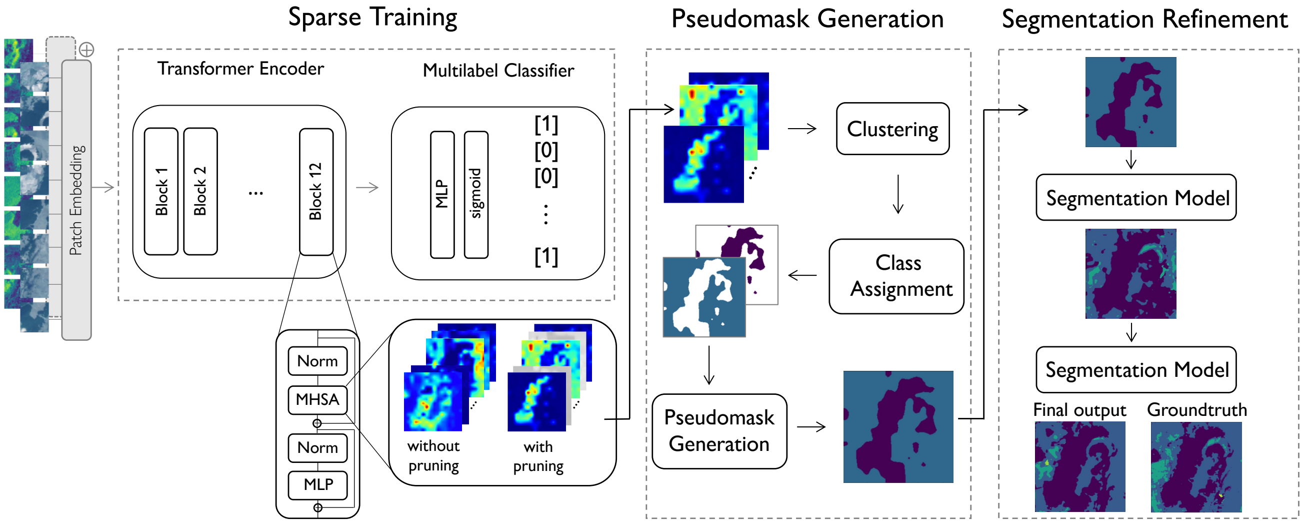

Figure 2. Our weakly-supervised semantic segmentation approach consists of a pipeline with three stages. In the first stage, we train a Transformer model for multi-label classification. During the training process, we enforce attention sparsity by adding learnable gating units to each head of the Multi-Head Self-Attention (MHSA). In the second stage, we generate pseudomasks by extracting attention maps from the last block of the Transformer, reshaping them and clustering them. Each cluster is assigned a label that will be used to create the corresponding pseudomask. In the final step, we refine the generated pseudomask through multiple stages of supervised training of a segmentation model, utilizing the pseudomasks generated in the previous iteration as supervision.

This is equivalent to turning off some of the connections in the network. On the other hand, structured pruning involves removing larger structures such as neurons, channels [13], filters, or attention heads [17].

Traditionally, pruning methods involve two steps. First, a dense network is trained, followed by pruning and fine-tuning. To avoid this process, several sparse training methods have been proposed. In these methods, network pruning takes place during training, and the model learns to optimize both the model weights and sparse connectivity simultaneously from scratch. Sparse training methods primarily focus on convolutional networks; the majority of these methods use unstructured sparsity, while only a handful discuss training convolutional networks with structured sparsity. Mocanu *et al.* [18] propose a prune and growth procedure called Sparse Evolutionary Training (SET), improving training with fixed sparse connectivity. Mostafa and Wang [19] extend this work and propose a dynamic parametrization method that adaptively reallocates free parameters across the network based on a simple heuristic during training. Cheng *et al.* [4] explore integrating sparsity in vision Transformers and propose a sparse and structured prune-and-grow self-attention mechanism. On a related note, Bazi *et al.* [3] propose a simple compression technique that gradually prunes the encoder's layers of a Transformer, for classifying remote sensing images.

In contrast to prior research, our approach involves following on Transformer pruning techniques that have been designed for natural language processing (NLP) [29] in order to sparsify the model during training, rather than gradually pruning and fine-tuning.

3. Dataset

Previous research has proven that combining multimodal remote sensing data can significantly improve the accuracy of land cover mapping by providing complementary spectral and structural information [8, 22].

Therefore, in this work, we use the DFC2020 dataset [23], which was constructed for the IEEE GRSS Data Fusion Contest 2020. This dataset contains synthetic aperture radar (SAR) observations from Sentinel 1 [26] and multispectral observations from Sentinel 2 [7], both of which are Earth observing satellites and part of the Copernicus program. The dataset consists of two sets of 986 and $5,128$ paired Sentinel 1 and Sentinel 2 observations, for training and testing respectively. Each image in the dataset has spatial dimensions of 256×256. 20% of the training set is reserved for validation.

In addition to the satellite imagery, the DFC2020 also provides dense (i.e., pixel-level) land cover annotations for the classes `Forest`, `Shrubland`, `Grassland`, `Wetland`, `Cropland`, `Urban`, `Barren` and `Water`, which are significantly unbalanced. These maps are used in two ways: (i) for generating image-level labels (only considering classes that cover $\geq 10\%$ of the image) for the training of our model and (ii) for assessing the accuracy of the generated pseudomasks.

4. Methods

Figure 2 illustrates our overall approach, which we detail in the following.

4.1. Transformer Architecture

In this work, we use a vision Transformer (ViT) [6]. Each input image $X \in \mathbb{R}^{H \times W \times C}$ (where H represents the height, W the width, and C the number of channels) must first be transformed into a sequence of flattened 2D patches $\mathrm{x}_p \in \mathbb{R}^{N \times P^2 \times C}$. Here, N corresponds to the total number of patches, and $P^2 = (P \times P)$ is the resolution of each patch. These patch-wise sequences are then linearly embedded and augmented with position embeddings before being encoded by a standard Transformer encoder. In classification tasks, a commonly adopted approach is to introduce a learnable ”classification token” `[CLS]` to the sequence of embedded patches.

To take advantage of our multimodal input, we use Early Summation [31]. Essentially, this involves adding the token embeddings from both modalities at each token position, prior to processing by the Transformer layers. Overall, Early Summation is a straightforward yet powerful way to enable multimodal interaction.

Each block contains two sub-layers: a multi-head self-attention (MHSA) layer and a multi-layer perceptron (MLP) [28]. To ensure efficient backpropagation of the gradient, both the MHSA and MLP layers employ residual connections. The MHSA mechanism uses scaled dot-product attention to operate on three inputs: query Q, key K, and value V. A Transformer with h heads produces one representation for (Q, K, V) per head, with each representation undergoing scaled dot-product attention. The resulting outputs are concatenated and then passed through a feed-forward layer. This can be expressed as:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{concat}_i(\mathrm{head}_i)W^O \tag{1}$$

With:

$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{(QW_i^Q)(KW_i^K)^T}{\sqrt{d_k}}VW_i^V\right) \tag{2}$$

where $W_i^Q, W_i^K, W_i^V, W_i^O$ are parameter matrices.

4.2. Sparse Training

A common observation across several works is that most of these heads are unnecessary and can be removed without any impact on performance. Our proposed solution is to introduce sparsity during training, which will encourage the heads to become more discriminative and less redundant. This should result in them carrying more meaningful information.

Building upon previous research in the field of NLP [29], we modify the vision Transformer architecture by incorporating gating units into each head. In this modification, scalar gates denoted as g_i are introduced and multiplied by the output of each head i. It is worth noting that these gates are specific to each head, resulting in unique values for each g_i. Equation 1 becomes then:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{concat}_i(g_i \cdot \mathrm{head}_i)W^O \tag{3}$$

The goal is not only to reduce the impact of less significant heads, but rather to disable them completely. To achieve this, L0 regularization is preferred, but since it is non-differentiable it and cannot be used as a regularization term in the objective function. To overcome this, a stochastic relaxation method is employed, using Hard Concrete distributions [15, 16]. These distributions are mixed, both discrete and continuous, and have a non-zero probability at 0 and 1. The sum of the probabilities of heads being non-zero can be used as a relaxation of the L0 norm.
The training objective is thus:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{bce} + \lambda\mathcal{L}_{reg} \\ &= \mathcal{L}_{bce} + \lambda\sum_i(1 - P(g_i = 0|\phi_i))\end{aligned} \tag{4}$$

where ϕ_i are the Hard Concrete distribution parameter and $\mathcal{L}_{bce}$ the binary cross-entropy loss for the multi-label classifier. By adjusting the coefficient λ in the training objective, we can generate models that preserve varying numbers of heads.

4.3. Mask Generation

As outlined in Section 4.1 the attention weights of a Transformer block are computed between the key (K) and the query (Q). The weights quantify how important the key is to the query. In a vision Transformer, the key and the query come from the same image, hence the weights determine which part of the image is important.

Our idea is thus to look at the self-attention of the `[CLS]` token on the heads of the last layer. By including the class token `[CLS]` in the input sequence, the self-attention mechanism is encouraged to distribute information among all the tokens in the sequence, and then combine their information into a representation that theoretically distinguishes between the various classes.

Clustering. The proposed pipeline is as follows: For each image $X \in \mathbb{R}^{H \times W \times C}$ we extract the representations from the `[CLS]` token ($Z \in \mathbb{R}^{1 \times \frac{H \times W}{P^2}}$) and reshape the attention scores to square image-like dimensions ($\mathbb{R}^{\frac{H}{P} \times \frac{W}{P}}$).

Table 1. Comparison of pseudomasks generated with our method with fully-supervised and weakly-supervised baselines, using two types of input: Sentinel-2 only (S2) and fusion of Sentinel-1 and Sentinel-2 (S1+S2). The first set of baselines are fully-supervised models trained on segmentation labels from the training set, utilizing UNets and ViT with segmentation head. The second set of baselines are weakly-supervised models, trained on image-level labels and generating pseudomasks with Grad-CAM. Our method has two setups: one without attention head pruning, and one with pruning. To estimate an upper-bound on the performance of our method, we report our results using k_{ref} (see Section 5.2). For all weakly-supervised scenarios, metrics are computed between the generated pseudomasks (without refinement) and the groundtruth. Per-class accuracies and mean Intersection over Union (bottom row only) are reported with their standard deviations from 5 runs. Per-class pixel-wise distribution in our training set is mentioned next to each class.

	Baselines						**Our Method**				
	Fully Supervised				Weakly Supervised		Weakly Supervised				
Class	UNet		ViT		ViT + Grad-CAM		w/o pruning		w/ pruning		w/ k_{ref}
	S2	S1+S2	S2	S1+S2	S2	S1+S2	S2	S1+S2	S2	S1+S2	S1+S2
Forest (9%)	**76 ± 1**	76 ± 2	72 ± 3	68 ± 1	34 ± 1	47 ± 3	51 ± 1	55 ± 3	59 ± 2	65 ± 1	79 ± 2
Shrubland (5%)	18 ± 1	11 ± 3	18 ± 3	8 ± 1	**34 ± 3**	33 ± 1	19 ± 2	21 ± 1	11 ± 1	18 ± 3	31 ± 2
Grassland (12%)	13 ± 3	**25 ± 1**	19 ± 3	20 ± 2	14 ± 1	11 ± 4	11 ± 3	15 ± 2	8 ± 1	19 ± 2	15 ± 1
Wetland (18%)	4 ± 1	7 ± 2	5 ± 2	5 ± 2	7 ± 1	8 ± 1	7 ± 1	10 ± 2	**11 ± 1**	6 ± 1	12 ± 2
Cropland (13%)	**62 ± 3**	52 ± 4	33 ± 1	39 ± 2	31 ± 4	38 ± 2	46 ± 5	44 ± 4	46 ± 4	48 ± 4	55 ± 3
Urban (5%)	38 ± 2	48 ± 2	46 ± 2	42 ± 2	52 ± 3	57 ± 2	59 ± 1	65 ± 3	**71 ± 2**	70 ± 4	70 ± 3
Barren (3%)	33 ± 2	30 ± 2	20 ± 1	21 ± 3	10 ± 1	18 ± 1	34 ± 1	40 ± 1	**47 ± 2**	43 ± 3	51 ± 2
Water (35%)	92 ± 1	95 ± 2	91 ± 1	93 ± 1	90 ± 1	92 ± 2	93 ± 2	94 ± 3	96 ± 3	**97 ± 2**	96 ± 2
Overall	**57 ± 2**	55 ± 1	47 ± 2	46 ± 2	43 ± 3	48 ± 1	51 ± 3	53 ± 2	57 ± 3	**57 ± 2**	62 ± 2
Average	42 ± 3	44 ± 1	38 ± 2	37 ± 2	34 ± 3	38 ± 1	40 ± 2	43 ± 1	44 ± 2	**46 ± 2**	51 ± 1
mIoU	34 ± 2	**37 ± 1**	28 ± 2	30 ± 3	28 ± 1	31 ± 1	32 ± 2	33 ± 1	34 ± 1	36 ± 1	40 ± 2

We refer to these as *attention maps*. The number of attention maps corresponds to the number of heads remaining (not pruned) after training and they ideally highlight different semantic information in the data. Nonetheless, there is a possibility that multiple heads emphasize similar information; as a result, these heads need to be clustered into C groups, where C corresponds to the number of labels predicted by the classification model. For this, we use a simple k-means algorithm with $k = C$, resulting in a single cluster per predicted class.

Cluster Assignment. A major problem is that the clusters give no information on their labels. To solve this issue, we propose to binarize the average image in each cluster and use it to mask the inputs to identify the corresponding class. Essentially, for each input image, we retain only pixels that correspond to positive values in each thresholded mask, and then run it through the trained classifier (forward pass) to determine the most likely label associated with it. We repeat this process for the C masks (for each cluster), until each cluster has been assigned to a different class.

Pseudomask Generation. After assigning each cluster to a specific class, we generate the multi-channel pseudomask by merging all of the class-level masks produced in the previous step.

Table 2. Results of the refinement process, after iteratively training a segmentation model (UNet), starting with the pseudomask generated with our method (compare to Table 1). We report the fully-supervised UNet for comparison. Results are shown after 1, 2 and 3 iterations of training, in terms of mean IoU and average pixel accuracy. Each iteration is trained for 10 epochs.

	UNet	Ours	1 it.	2 it.	3 it.
Accuracy	44.3	46.1	46.5	46.6	**47.1**
mIoU	37.2	36.4	38.3	38.1	**39.2**

4.4. Segmentation Training

After generating the pseudomasks, the last step in the process is training a segmentation model that will refine these masks. To do this, we use a standard UNet [21] that takes as input an early concatenation of the two modalities (Sentinel 1 and Sentinel 2), and uses as target the pseudomasks. This training is fully-supervised and is repeated iteratively.

5. Experiments

We perform a series of experiments to evaluate the performance of our weakly-supervised approach on the DFC2020 dataset. We train our models on approximately 800 multimodal samples as specified in Section 3 and evaluate them on around $5,000$ multimodal samples. To em-

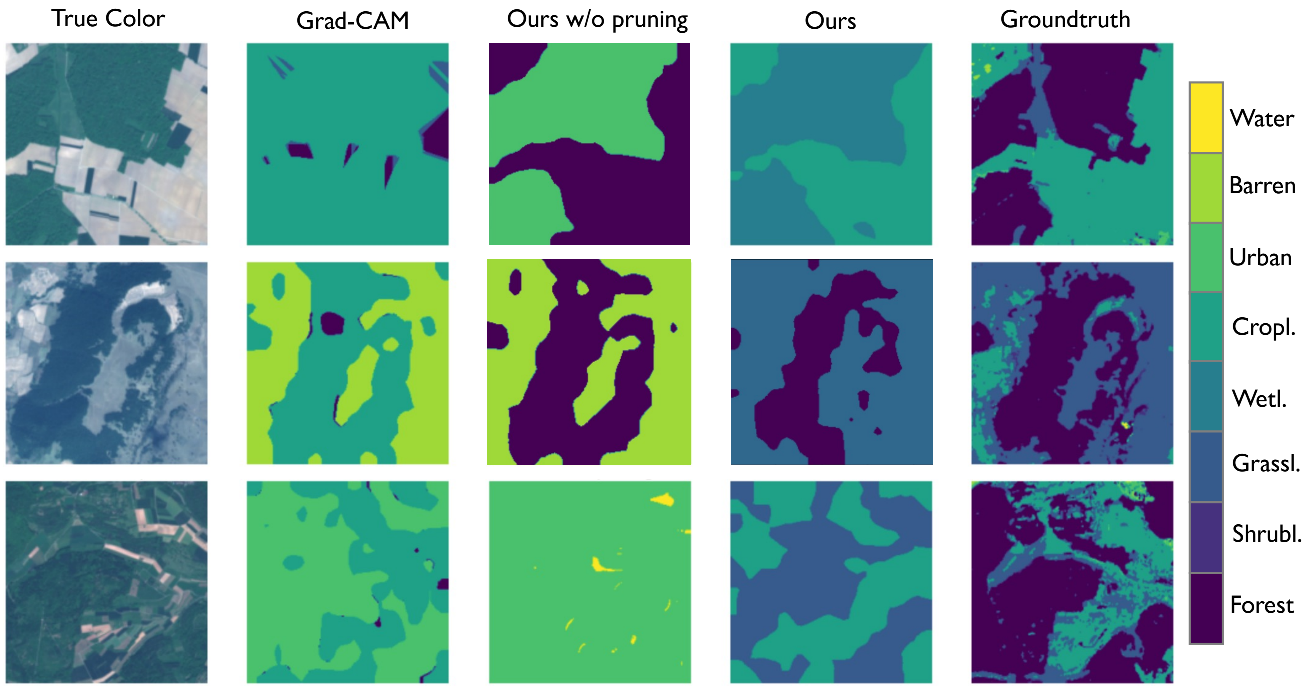

Figure 3. Qualitative comparison of results for 3 different regions, using weak supervision. Results from left to right: Sentinel-2 true color (RGB), pseudomask generated from Grad-CAM, pseudomask generated using our method, pseudomask generated using our method with pruning, and finally the groundtruth.

phasize the advantage of multimodality, we also train our models with only the Sentinel-2 modality and compare the performances.

5.1. Implementation Details

Our method outlined in Section 4 and Figure 2 is implemented using the Pytorch framework [20]. The model is trained on the training set, and a grid search is used to determine the hyper-parameters based on a validation set. The AdamW optimizer [14] was used during training, using $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The learning rate was set to 10^{-4}. We used a cosine learning rate schedule with a linear warmup. We trained our model for 300 epochs with a batch size of 32 using 1 Nvidia Tesla V100 GPU.

Regarding the ViT's hyperparameters, we use patch sizes of 14×14, and a depth (nb. of blocks) of 12. Throughout this work, we use 16 heads and a λ of 0.01 (Equation 4).

5.2. Baselines

We compare our approach with traditional fully-supervised segmentation methods as well as other weakly-supervised methods.

Fully Supervised For comparison, we use a UNet [21] and a ViT encoder-based approach, to which we append a segmentation head as a replacement for the classification head. The segmentation head consists of convolution and interpolation layers to obtain the desired spatial and channel dimensions: $H \times W \times C$, where H and W are the height and width of the input image respectively, and C the total number of classes.

Weakly Supervised As our baseline, we use Grad-CAM (Gradient-weighted Class Activation Mapping) [24]. Since the model is trained exclusively on image-level labels, it can only provide information about the presence or absence of a particular object, not its location. Grad-CAM addresses this limitation by generating a heatmap that highlights the regions of the image that the model used to make its prediction. Each activation class is linked to a specific output class, thus allowing to determine the importance of each pixel in the final decision of the model. This importance is reflected by the intensity of the pixels which is increased or decreased according to the contribution of each pixel to the class concerned. The distinct maps are first binarized and then combined together to create the pseudomask.

We compare our approach to the two fully-supervised baselines and the weakly-supervised one. Evaluation metrics (pixel-accuracies and mean Intersection over Union (mIoU)) are shown in Table 1. Our weakly-supervised approach with pruning demonstrates the best average per-

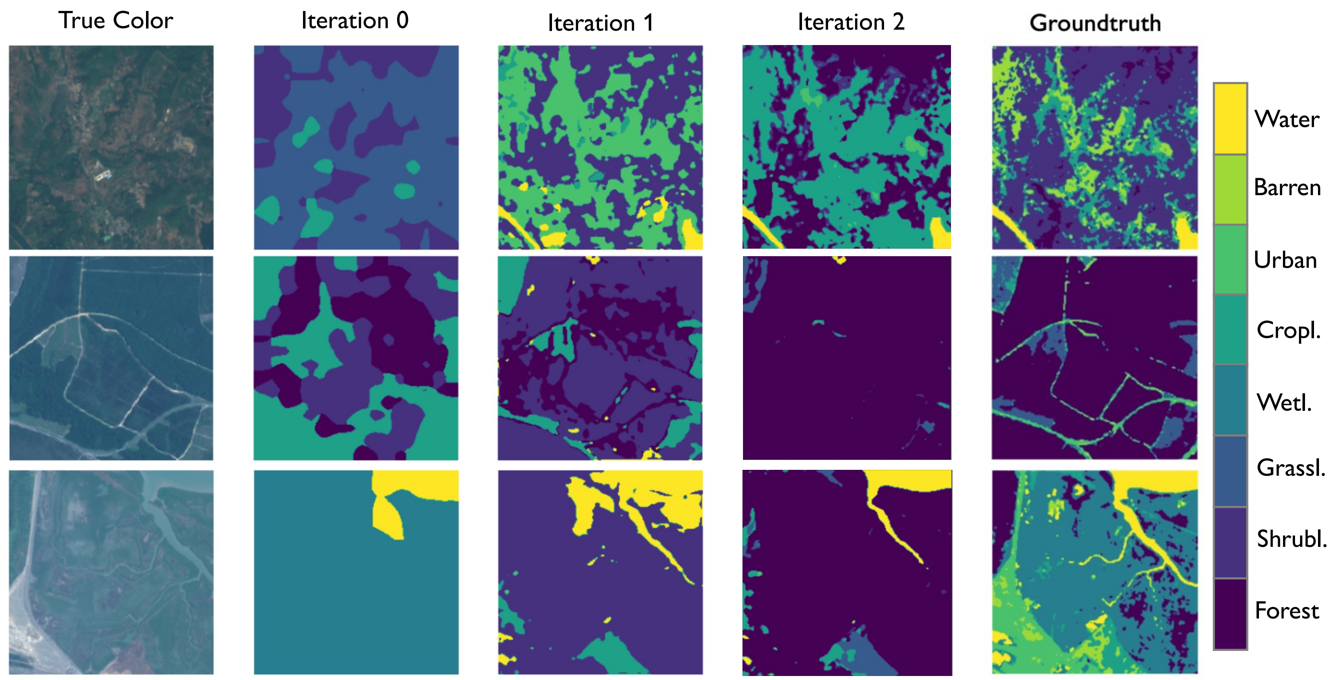

Figure 4. Qualitative comparison of the iterative refinement process of pseudomasks. The initial pseudomask (iteration 0) undergoes 1 and 2 iterations of refinement, which consists in training a segmentation model in a supervised way by using as supervision the masks generated by the previous iteration.

formance for multimodal input in terms of pixel-accuracy, with an increase of 2 and 9 percentage points compared to the fully-supervised UNet and ViT baselines, respectively. Although our approach performs almost similarly to the UNet with regards to mIoU, it does not rely on any dense label as supervision. Additionally, our method outperforms the weakly-supervised baseline that uses Grad-CAM, with an improvement of 8 and 5 percentage points in pixel accuracy and mIoU, respectively.

In Table 1, we take a further step in evaluating our approach by utilizing the actual number of clusters (denoted as k_{ref}, in reference to the k of the k-means) in our pipeline, which is equivalent to the number of 'true' labels per image. This is in contrast to using the number of labels predicted by the Transformer as the value of k. By comparing the results using k_{ref}, we can estimate an upper-bound of our method, assuming the initial classifier was close to perfect. The results demonstrate that by using k_{ref}, we could achieve an improvement of approximately 5 percentage points in terms of mIoU and accuracy.

5.3. Effect of Sparsity

We examine the impact of enforcing sparsity during training, in two stages. Firstly, we evaluate the Transformer's performance on the multi-label classification task. We vary the λ parameter in the objective function (shown in Equation 4). The resulting models have different numbers of heads retained, as presented in Table 3. The larger λ, the sparser the network becomes. We note that for the setup with 16 heads, removing 72% of attention heads (using $\lambda = 0.01$) results in an accuracy drop of the multi-label classifier of less than 1% compared to the model with the full number of heads. Secondly, we validate the use of sparsity during the mask generation process. Table 1 shows that our weakly-supervised approach with sparsity enforcement produces similar mIoU values as the UNet baseline trained using fully-supervised learning. Additionally, our method achieves the best pixel accuracy compared to all other methods.

Table 3. Effects of λ (Equation 4) on sparsity rate (percentage of pruned heads) and multi-label accuracy, for different numbers of heads h: 8, 16 and 32.

		$\lambda = 0$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$
h = 8	Sparsity rate	0	44	71	78
	Accuracy	72.8	69.1	64.2	65.7
h = 16	Sparsity rate	0	54	72	88
	Accuracy	79.4	79.6	78.9	74.8
h = 32	Sparsity rate	0	62	72	89
	Accuracy	79.9	78.8	76.1	75.7

5.4. Qualitative Comparison

We conduct a visual inspection of the results to supplement our numerical evaluation of the approach. Figure 3 compares our results to the baseline. The pseudomasks generated by Grad-CAM are presented first, followed by the ones generated using our approach without any pruning enforcement, and finally those generated with pruning. We observe that our method with pruning produces the most accurate pseudomasks in terms of shapes and classes.

5.5. Iterative Pseudomask Refinement

The refinement process starts with the initial pseudomask and involves multiple stages of training a segmentation model in a supervised manner, utilizing the masks generated in the previous iteration as supervision. We present the evaluation metrics for each stage in Table 2. Our results show that the values improve gradually after each iteration. Additionally, we conduct a visual inspection of the refinement process in Figure 4. We begin by generating a pseudomask using our approach with pruning, which we denote as Iteration 0. Subsequently, the pseudomask undergoes 2 iterations of refinement. We observe that after each iteration the details are refined, evolving from coarse at Iteration 0 to fine-grained at the end, and the shapes become increasingly similar to those found in the ground truth.

6. Discussion

Our work highlights the benefits of enforcing sparsity in vision Transformer training for weakly-supervised semantic segmentation. By following our approach, we achieve comparable performance to standard fully-supervised training, without the requirement of fine-grained labels.

First, on the data side, it is interesting to note that there is a significant advantage in using both modalities (Sentinel 1 and Sentinel 2) together instead of the Sentinel 2 modality alone (see Table 1). This is because they offer complementary information, which enhances the initial classification task. Second, on the model side, we find that almost 75% of the heads in the MHSA can be pruned without any significant impact on the Transformer's classification performance, as presented in Table 3. This means that not all heads are necessary for the Transformer to learn effectively. Interestingly, we observe that training a Transformer with a reduced number of heads ($h = 8$) does not lead to equal results. Thus, it is more beneficial to start with a larger number of heads and prune them while training. Furthermore, pruning the Transformer has a significant impact on the process of generating pseudomasks. Our approach heavily relies on the learned representations, and thus, removing unnecessary and redundant units leads to more accurate and less noisy pseudomasks. This improvement is observed in Table 1, where our pruning-based approach outperforms the un-pruned one by 3 percentage points in pixel accuracy and mIoU. The advantages of pruning can also be observed in a visual comparison, as shown in Figure 3. By qualitatively evaluating the pseudomasks generated by our approach with and without pruning, we can observe that pruning improves the level of details of the different elements in the pseudomasks, thereby increasing the similarity to those details present in the ground truth, compared to any other method. It should be noted that our approach has inherent uncertainties associated with the cluster label assignment, which is a common problem in unsupervised learning. This can be observed in Figures 3 and 4, where the shapes in the pseudomasks are accurate, but the assigned labels are incorrect.

Regarding the iterative process of pseudomask refinement, it is worth noting that after each iteration, the evaluation metrics improve, as illustrated in Table 2. This is also visually observable in Figure 4, where each iteration refines the details further, and sometimes corrects errors from previous ones. However, it is crucial to avoid training the refinement model (UNet) excessively during each iteration, as doing so may result in overfitting of the pseudomasks. Instead, our goal is to utilize the model to learn a certain level of consistency.

On a final note, our method can reduce the need for fine-grained labeled data by generating high-quality pseudomasks that, when refined iteratively, yield results akin to those obtained via human-annotated ground truth. Our approach is particularly effective even on small-scale datasets. However, it is worth noting that, while our approach outperforms both qualitatively and quantitatively the standard weakly-supervised one using Grad-CAM, it requires slightly more time to generate the pseudomasks.

7. Conclusion

This work demonstrates the potential to create pseudomasks for segmentation using weak supervision by leveraging representations learned by Transformers. The pipeline consists of three stages. In the first stage, the Transformer is trained on multi-label classification. To achieve a compact and informative representation, we add learnable gating units to each head of the MHSA during training, which enforces sparsity and retains only the most meaningful heads. In the second stage, we generate the pseudomask by looking at the `[CLS]` token on the un-pruned heads of the final Transformer layer. In the third stage, we refine the pseudomask through multiple supervised training iterations using the pseudomask from the previous iteration as supervision. Our experiments demonstrate that a significant number of heads in a vision Transformer can be pruned without affecting its performance. By combining our weakly-supervised approach with sparse attention, we achieve comparable or even slightly superior results to fully-supervised baselines, without requiring fine-grained labeled data.

References

[1] Steve Ahlswede, Thekke Madam Nimisha, Christian Schulz, Birgit Kleinschmit, and Begüm Demir. Weakly supervised semantic segmentation of remote sensing images for tree species classification based on explanation methods. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 4847–4850, 2022. 2

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. 2

[3] Yakoub Bazi, Laila Bashmal, Mohamad Mahmoud Al Rahhal, Reham Al Dayil, and Naif Al Ajlan. Vision transformers for remote sensing image classification. *Remote. Sens.*, 13:516, 2021. 3

[4] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. In *Neural Information Processing Systems*, 2021. 3

[5] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xiansheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 959–968, 2022. 1, 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 2, 4

[7] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012. 3

[8] Luis Gómez-Chova, Devis Tuia, Gabriele Moser, and Gustau Camps-Valls. Multimodal classification of remote sensing images: A review and future directions. *Proceedings of the IEEE*, 103:1560–1584, 2015. 3

[9] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *arXiv: Computer Vision and Pattern Recognition*, 2016. 2

[10] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. *ArXiv*, abs/1506.02626, 2015. 2

[11] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1665–1674, 2017. 2

[12] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, 2016. 1, 2

[13] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017. 3

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 6

[15] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l0 regularization. *ArXiv*, abs/1712.01312, 2017. 4

[16] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ArXiv*, abs/1611.00712, 2016. 4

[17] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Neural Information Processing Systems*, 2019. 2, 3

[18] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9, 2018. 3

[19] Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. *ArXiv*, abs/1902.05967, 2019. 3

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019. 6

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. 5, 6

[22] Linus Scheibenreif, Joelle Hanna, Michael Mommert, and Damian Borth. Self-supervised vision transformers for land-cover segmentation and classification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1421–1430, 2022. 3

[23] Michael Schmitt, Lloyd Hughes, Pedram Ghamisi, Naoto Yokoya, and Ronny Hänsch. 2020 ieee grss data fusion contest, 2019. 3

[24] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2016. 6

[25] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3131–3140, 2019. 1, 2

[26] Ramon Torres, Paul Snoeij, Dirk Geudtner, David Bibby, Malcolm Davidson, Evert Attema, Pierre Potin, BjÖrn Rommen, Nicolas Floury, Mike Brown, et al. Gmes sentinel-1 mission. *Remote sensing of environment*, 120:9–24, 2012. 3

[27] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2687–2697, 2022. 2

[28] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 4

[29] Elena Voita, David Talbot, F. Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *57th Conference of the Association for Computational Linguistics*, pages 5797–5808, 2019. 2, 3, 4

[30] Yude Wang, Jie Zhang, Meina Kan, S. Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12272–12281, 2020. 2

[31] Peng Xu, Xiatian Zhu, and David A. Clifton. Multimodal learning with transformers: A survey. *ArXiv*, abs/2206.06488, 2022. 4

[32] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. 2