Layer Optimized Spatial Spectral Masked Autoencoder for Semantic Segmentation of Hyperspectral Imagery

Aaron Perez, PhD Student
Dept. of Electrical and Computer Engineering,
University of Houston, Houston, TX, USA
asperez@cougarnet.uh.edu

Saurabh Prasad, PhD
Dept. of Electrical and Computer Engineering,
University of Houston, Houston, TX, USA
saurabh.prasad@ieee.org

Abstract

Hyperspectral imaging (HSI) captures detailed spectral data across numerous contiguous bands, offering critical insights for applications such as environmental monitoring, agriculture, and urban planning. However, the high dimensionality of HSI data poses significant challenges for traditional deep learning models, necessitating more efficient solutions. In this paper, we propose the Layer-Optimized Spatial-Spectral Transformer (LO-SST), a refined version of the Spatial-Spectral Transformer (SST) that incorporates structured layer pruning to reduce computational complexity while maintaining robust performance. LO-SST leverages self-supervised pretraining with a Masked Autoencoder (MAE) framework, enabling the model to effectively learn spatial and spectral dependencies even in scenarios with limited labeled data. The use of separate spatial and spectral positional embeddings further enhances the model's ability to capture intricate relationships within hyperspectral data. Our experiments show that LO-SST achieves competitive segmentation accuracy while significantly reducing computational demands compared to traditional models. The effectiveness of random masking over alternative strategies during pretraining is also demonstrated, underscoring its ability to preserve critical image features. These results highlight the potential of LO-SST as an efficient and scalable solution for hyperspectral image segmentation, particularly in resource-constrained applications.

1. Introduction

Hyperspectral imaging (HSI) provides critical spectral data for applications such as environmental monitoring, agriculture, and urban planning [2, 19]. However, the high dimensionality of HSI data and the scarcity of labeled training data pose significant challenges for deep learning methods, particularly in scenarios with limited computational resources.

The objective of this research is to develop an efficient model for hyperspectral image segmentation that performs well in scenarios with limited labeled training data and computational resources. To achieve this, we propose the **Layer-Optimized Spatial-Spectral Transformer (LO-SST)**, a model that balances performance and efficiency through key innovations.

LO-SST employs layer pruning based on magnitude importance scores to reduce computational complexity while retaining robust performance. It incorporates separate 2D and 1D sinusoidal positional embeddings for spatial and spectral dimensions, capturing dependencies essential for hyperspectral data analysis. We adapt a Masked Autoencoder (MAE) with LO-SST, using 75% masking for pretraining, which is particularly important in situations with limited labeled data.

Our experiments evaluate the performance of LO-SST with various segmentation heads and masking strategies. The results demonstrate that LO-SST delivers competitive segmentation accuracy while reducing computational complexity. Key contributions of this work include:

1. **Efficient layer optimization SST:** Introducing pruning strategies to reduce computational complexity while maintaining robust performance.

2. **Enhanced spatial spectral embeddings:** Utilizing distinct 2D and 1D sinusoidal positional embeddings to enhance spatial and spectral feature representation. These embeddings are then concatenated and added to 3D hyperspectral patch embeddings.

3. **Comprehensive evaluation:** Analyzing segmentation heads and masking strategies for robust performance in low-data scenarios.

4. **Comparison of Pre-Trained Fine-Tuning and Full Model Tuning:** Evaluating the impact of fine-tuning pre-trained models versus retraining all components of the pre-trained model for hyperspectral segmentation.

This work highlights the potential of layer based optimization of transformer-based networks as a scalable, resource-efficient solution for hyperspectral image segmentation.

2. Related Work

2.1. Deep Learning for Hyperspectral Imaging

Hyperspectral imaging (HSI) captures spectral data across multiple bands, enabling detailed remote sensing and vision tasks [17]. While convolutional neural networks (CNNs) are commonly used for classification, their fixed receptive fields limit their ability to capture long-range dependencies, a critical aspect for segmentation tasks [2, 19]. 3D CNNs attempt to address this by analyzing both spectral and spatial dimensions, but their computational complexity and limited contextual understanding remain challenges [10].

2.2. Vision and Spatial-Spectral Transformers (SST)

Vision Transformers (ViTs), introduced in 2020, adapt the self-attention mechanisms from natural language processing to visual data by dividing images into fixed-size patches, transforming them into tokens with positional embeddings. These tokens are processed by a transformer encoder, enabling ViTs to capture both local and global relationships effectively. This architecture has proven highly successful in tasks like image classification and segmentation [8, 15].

Building on the success of ViTs, Spatial-Spectral Transformers (SSTs) have been developed to meet the unique demands of hyperspectral imaging. SSTs extend the self-attention mechanism to encode hyperspectral data as spatial-spectral tokens, capturing both local structures and long-range dependencies across spectral bands. By addressing challenges such as high dimensionality and spectral redundancy, SSTs have achieved state-of-the-art performance in pixel-wise classification, segmentation, and super-resolution, setting new standards for hyperspectral image analysis [4, 13, 20, 25].

2.3. Vision Transformer Layer Pruning

Vision Transformer (ViT) layer pruning is a key technique for reducing computational and memory demands while preserving performance [16]. Studies, such as one by Chen *et al*, show that pruning can achieve high sparsity with minimal accuracy loss [3]. Advanced methods like X-Pruner [23] and Hessian-aware pruning [22] further optimize this process by considering layer interdependencies and structural pruning.

Magnitude-based pruning, originally developed for traditional neural networks , removes low-magnitude parameters, such as neurons, to simplify models while retaining performance. This approach relies on distribution-free uncertainty measures to identify and eliminate less significant components [6], resulting in more efficient and sparse architectures [1]. While primarily used in neural networks, these principles could be adapted to ViTs by targeting parameters like attention heads, projection layers, and MLP blocks. Applying this method to ViTs could create simpler, more efficient models, enabling their deployment on resource-constrained devices or in latency-sensitive scenarios.

2.4. Masked Image Modeling in Hyperspectral Imaging

Masked image modeling (MIM) is a self-supervised learning technique designed to address challenges in hyperspectral imaging, such as high dimensionality and limited labeled data. By masking portions of the input and training the model to reconstruct the missing regions, masked autoencoders (MAEs) enable feature learning without requiring annotations, making MIM particularly valuable in data-scarce domains [9, 24].

MAEs enhance efficiency by encoding only unmasked data, reducing computational overhead while encouraging the learning of contextually meaningful features. The decoder reconstructs the masked regions, improving the model's generalization and ability to extract robust representations [21]. This dual focus on efficiency and contextual learning makes MAEs well-suited for hyperspectral imaging.

When paired with Vision Transformers (ViTs) or Spatial-Spectral Transformers (SSTs), MAEs effectively capture long-range dependencies and spatial-spectral correlations. Their self-supervised design enables state-of-the-art performance in classification and segmentation, unlocking the potential of hyperspectral data [18].

3. Methods

3.1. Transformer Architecture

In this work, we build upon the existing Spatial-Spectral Transformer (SST) framework [18], which refines the Vision Transformer (ViT) [8] for hyperspectral data processing. SST addresses challenges of high-dimensional hyperspectral data by dividing the input $x \in \mathbb{R}^{(h \times w \times c)}$ into spatial-spectral patches of size $p_h \times p_w \times p_c$, resulting in $n = \left(\frac{h}{p_h}\right) \times \left(\frac{w}{p_w}\right) \times \left(\frac{c}{p_c}\right)$ patches. This allows simultaneous interpretation of spatial and spectral features and incorporates **blockwise spectral embedding**, where each spectral block defined by $\frac{c}{p_c}$ is assigned a distinct linear embedding to capture spectral diversity [7].

To further enhance the SST framework, we utilize both spatial and spectral positional embeddings. Specifically:

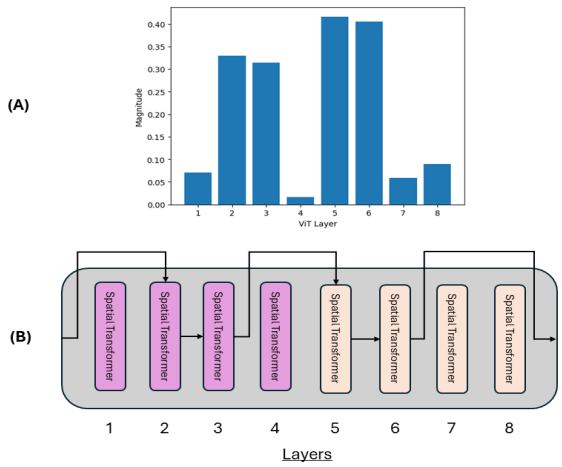

Figure 1. Illustration of the Layer Optimization process in the LO-SST framework. **(A)** Layers 1 and 4 (spatial) and 7 and 8 (spectral) are identified as low-magnitude and pruned. **(B)** Lines indicate data flow, skipping the pruned transformers to simplify the model and reduce computational complexity.

- A 2D sinusoidal embedding is applied to encode spatial positions, capturing spatial relationships effectively.
- A 1D sinusoidal embedding is applied to encode spectral positions, modeling dependencies across spectral bands.

These positional embeddings are concatenated and added to the patch embeddings, enabling the model to capture relationships across both spatial and spectral dimensions more effectively:

$$\begin{aligned} \text{PE}_{\text{pos},2i} &= \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \\ \text{PE}_{\text{pos},2i+1} &= \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \end{aligned} \tag{1}$$

where *pos* represents the position index, *i* is the dimension index, and *d* is the dimensionality of the model's input.

Finally, the enhanced patch embeddings are processed by both spatial and spectral transformers, extending self-attention across both dimensions [12, 18]. By employing spatial and spectral positional embeddings, we further improve SST's ability to model complex relationships within hyperspectral data.

3.2. Layer Optimized Spatial-Spectral Transformer (LO-SST)

We introduce the **Layer Optimized Spatial-Spectral Transformer (LO-SST)**, which incorporates structured layer pruning into Spatial-Spectral Transformers (SST) to reduce model complexity while maintaining performance. Layers are pruned based on their magnitude importance, calculated as:

$$\text{Layer Importance}_{L_i} = \frac{1}{P}\sum_{p=1}^{P}|\theta_p|, \tag{2}$$

where P is the total number of parameters in layer L_i, and $|\theta_p|$ is the absolute value of parameter θ_p. The layers with lowest importance are pruned, with parameters $theta$ in pruned layers zeroed out: $theta \rightarrow 0$.

Pruning is applied every 10 epochs, allowing the model to adapt iteratively. The original parameters are stored for potential restoration during fine-tuning. This structured pruning simplifies the model and aligns with the modularity of transformers, reducing complexity without sacrificing performance. The workflow of the LO-SST pruning process is illustrated in Figure 1.

3.3. Layer Optimized Masked SST Autoencoder (LO-MSST)

We extend the LO-SST into a masked image modeling framework, referred to as the **Layer Optimized Masked SST Autoencoder (LO-MSST)**. The hyperspectral cube is divided into 3D spatial-spectral patches of size $p_h \times p_w \times p_c$, enriched with positional embeddings (2D for spatial and 1D for spectral dimensions). Following this, self-supervised pre-training is undertaken using the entire hyperspectral image cube, where 75% of patches are masked. We study two masking strategies:

- **Random Masking**: This approach selects patches randomly by assigning scores sampled from a uniform distribution and masks patches based on the masking ratio [9, 18]. This is akin to creating random masks across both spatial and spectral dimensions.
- **Tube Masking**: In this method, spatial patches are selected based on random scores, and all spectral blocks at those locations are masked, creating a "tube" of masked data [14].

The unmasked patches are processed sequentially by spatial and spectral transformers. A lightweight decoder reconstructs the patches using mean squared error (MSE) loss, ensuring robust spatial-spectral feature learning. Layer optimization during training further enhances efficiency without compromising performance. Figure 2 illustrates the workflow.

3.4. Segmentation Head Evaluation

The segmentation process for the LO-MSST begins by leveraging the pre-trained LO-MSST encoder. The encoder's bottleneck, which provides a compact representa-

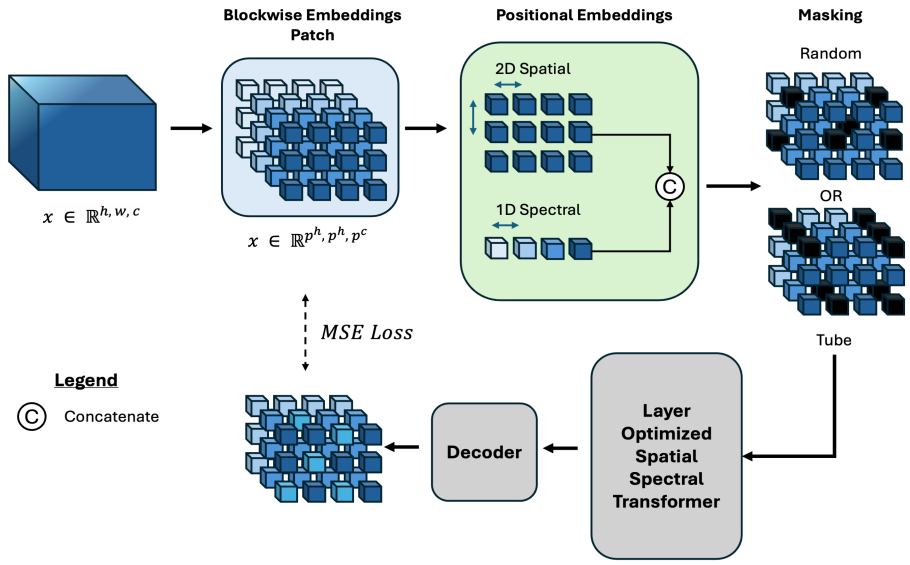

Figure 2. Workflow of the Layer-Optimized Masked Spatial-Spectral Transformer (LO-MSST), where a hyperspectral cube is divided into smaller 3D patches. Each patch undergoes block-wise spatial-spectral embedding, with positional embeddings applied both spatially and spectrally and subsequently concatenated. The process includes masking, LO-SST transformation, and final reconstruction through a lightweight decoder.

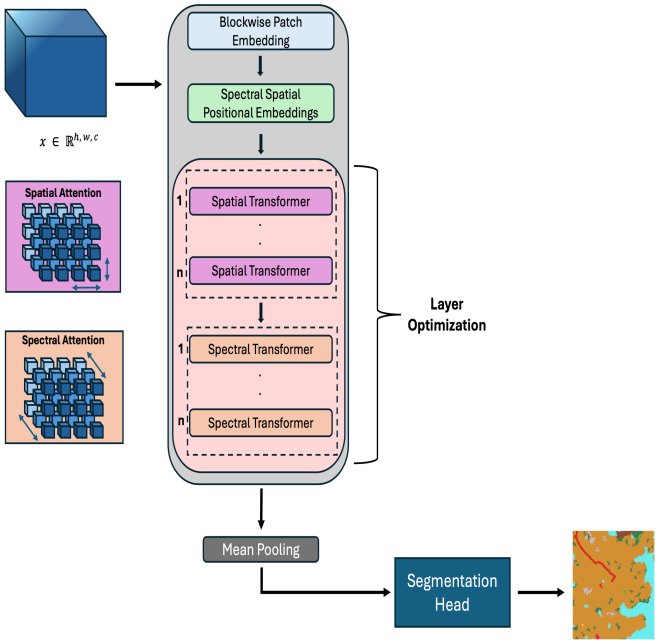

Figure 3. Architecture of the Layer-Optimized Spatial-Spectral Transformer (LO-SST) for hyperspectral segmentation.

tion of the input hyperspectral data, serves as input to the segmentation head. The process involves three stages:

- **Fine-Tuning the Pre-Trained Model:** Train only the segmentation head while keeping the encoder frozen.
- **Fine-Tuning the Full Model:** Unfreeze the encoder and jointly optimize both the encoder and segmentation head.
- **Model Comparison:** Compare our segmentation models with state-of-the-art models like Mask2Former

[5] and a standard SST without layer optimization to evaluate the efficiency and performance gains of LO-MSST.

To further evaluate segmentation performance of the LO-MSST encoder, we investigate three different architectures for the segmentation head:

- **Linear Head:** A single fully connected layer directly maps bottleneck features to pixel-wise predictions. The output is reshaped to match the input image dimensions, making it a lightweight and efficient architecture suitable for straightforward segmentation tasks.
- **CNN-Decoder:** The decoder includes three up-convolution layers (kernel size = 2, stride = 2) for upsampling and three blocks of two consecutive convolutional layers (kernel size = 3, padding = 1). The first layer extracts patterns, and the second refines them. This structure increases the resolution from 16×16 to 128×128 across nine layers.
- **UNet:** Building on the CNN decoder, UNet uses six up-convolution layers and ten blocks of two convolutional layers each. Skip connections from the encoder preserve details, progressively increasing resolution from 16×16 to 128×128.

The architecture of the LO-SST hyperspectral segmentation is illustrated in Figure 3.

4. Dataset

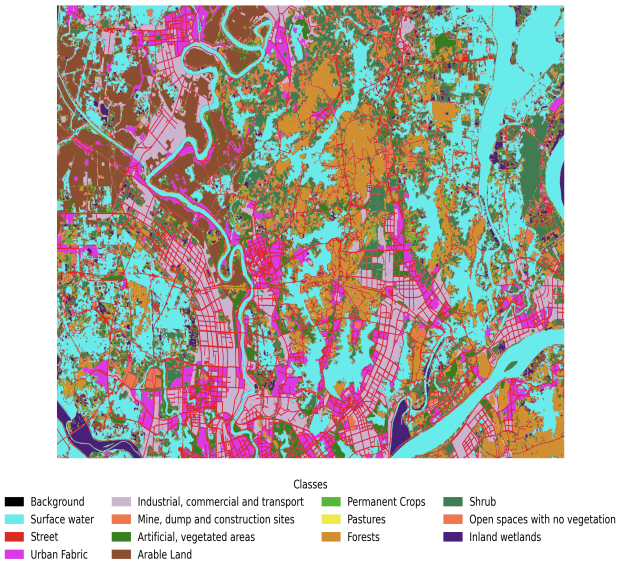

Figure 4. Class labels in the RS cross-city dataset, where each color represents a distinct land cover type.

The Remote Sensing (RS) cross-city dataset focuses on semantic segmentation in urban regions. It utilizes hyperspectral imagery from the Gaofen-5 satellite (CRESDA) for Wuhan, providing high-resolution data with dimensions $116 \times 2890 \times 2075$, where 116 corresponds to spectral bands after preprocessing. The original 300 spectral bands were reduced to 116 using water vapor absorption correction and Savitzky-Golay filtering [11].

Ground truth labels were derived from OpenStreetMap's LULC platform, ensuring accurate land use and land cover (LULC) annotations. A visualization of the class labels is shown in Figure 4, and Table 1 lists the class categories used for segmentation

The hyperspectral image was normalized and initially divided into patches of size $116 \times 128 \times 128$ for processing. After applying data augmentation techniques such as rotations of $0°$, $90°$, and $270°$, a total of 1,408 patches were generated. The dataset was then split into an 80/20 ratio for training and testing.

Class ID	Description
0	Background
1	Surface water
2	Street
3	Urban Fabric
4	Industrial, commercial and transport
5	Mine, dump and construction sites
6	Artificial, vegetated areas
7	Arable Land
8	Permanent Crops
9	Pastures
10	Forests
11	Shrub
12	Open spaces with no vegetation
13	Inland wetlands

Table 1. Land cover class descriptions for the RS cross-city dataset.

5. Experiments

5.1. Masked Pre-Training

We developed eight LO-MSST models with two configurations: **Base** and **Large**. These configurations, summarized in Table 2, are optimized for hyperspectral data using specific spatial and spectral patch sizes. During training, the $116 \times 128 \times 128$ hyperspectral image was divided into $4 \times 8 \times 8$ 3D cubes for input processing. Random masking and tube masking strategies were explored using the Base configuration.

The **Base** model uses an embedding dimension of 48, 4 transformer layers, and 4 attention heads, while the **Large**

Table 2. Comparison of Base and Large Configurations for LO-SST Models.

Parameter	Base	Large
Embedding Dimension	48	84
Depth (Layers)	8	12
Number of Heads	8	12
Patch Size (Spatial & Spectral)	$8 \times 8, 4$	$8 \times 8, 4$
Layer Prunning	0, 2, 4, 6	0, 2, 4, 6

model uses an embedding dimension of 84, 6 transformer layers, and 6 attention heads. Both configurations adopt a spatial patch size of 8×8 and a spectral patch size of 4. During pre-training, 75% of pixels were masked, and the model was trained for 100 epochs using MSE loss to enforce robust feature learning.

Cluster limitations constrained parameter count to the Large configuration, requiring careful architectural tuning for computational feasibility. To evaluate the layer optimization strategy, we froze 0, 2, 4, and 6 layers and measured parameter reduction, as shown in Table 3. These experiments demonstrate the adaptability of the LO-MSST framework under varying constraints and masking strategies.

Table 3. Layer Pruning, Parameters Removed, and Remaining Parameters in LO-SST Models. The total number of parameters removed is calculated as $68064 \times$ layers pruned.

Layers Pruned	Removed	Remaining-Large	Remaining-Base
0	0	8,988,679	8,471,774
2	136,128	8,852,551	8,335,646
4	272,256	8,716,423	8,199,518
6	408,384	8,580,295	8,063,390

5.2. Segmentation Evaluation

The segmentation process for the LO-MSST is conducted in three phases. Each phase involves training for 200 epochs using cross-entropy loss.

Phase 1: Fine-Tuning the Pre-Trained Model In this phase, the pre-trained LO-MSST encoder remains frozen while only the segmentation head is trained, utilizing the encoder's bottleneck as input. Three segmentation heads are evaluated: Linear Head, CNN Decoder, and UNet.

Phase 2: Fine-Tuning the Full Model In this phase, the pre-trained LO-MSST encoder is unfrozen, and all parameters are fine-tuned using 30% of the training dataset.

Phase 3: Model Comparisons In this phase, we develop three models based on Mask2Former, vanilla Base SST, and vanilla Large SST, training each using the full dataset.

Table 4. Segmentation Head Experiment Results on pretrained models. Pixel Accuracy is shown for different segmentation heads and models with base SST.

Model	Head	Pixel Acc. (%)
B_(LO-SST)_0	Linear	46.51
B_(LO-SST)_0	CNN-Decoder	53.50
B_(LO-SST)_0	UNet	**53.75**
L_(LO-SST)_0	Linear	35.10
L_(LO-SST)_0	CNN-Decoder	**40.06**
L_(LO-SST)_0	UNet	30.68

6. Results

6.1. Masked Self-Supervised Pre-Training

The results of the Masked Autoencoder (MAE) experiments demonstrate that even with 75% of the pixels masked, the base model with random masking produces the most visually satisfactory outputs. As shown in Figure 5, the base model with random masking effectively reconstructs fine details, such as the river, preserving edges and maintaining critical image properties. In contrast, base tube masking results in over-smoothed reconstructions, losing sharpness and important structural details.

When comparing the base model with random masking and 2 layers pruned versus 6 layers pruned, the visualization of the 6-layer pruned model looks better, with clearer details of the river and the curves the land makes.

The large model, on the other hand, generated pixelated outputs that were not distinguishable from the original image, likely due to its increased size and insufficient training epochs to converge effectively. This is not show in the figure.

These observations highlight that **random masking** in the base model provides superior reconstruction quality, while also being more computationally efficient. Moreover, magnitude based layer pruning enhances computational speed without compromising reconstruction quality.

6.2. Segmentation

6.2.1 Fine-Tuning Performance of Segmentation Heads

For our experiments, the first letter **B** denotes the base model and **L** denotes the large model. The value at the end indicates the number of layers pruned.

The performance of three segmentation heads integrated with the B-(LO-SST) and L-(LO-SST)is shown in Table 4. The Linear Head provides a baseline accuracy of 46.51%, while the CNN-Decoder improves pixel accuracy to 53.50%. The UNet achieves the highest accuracy of

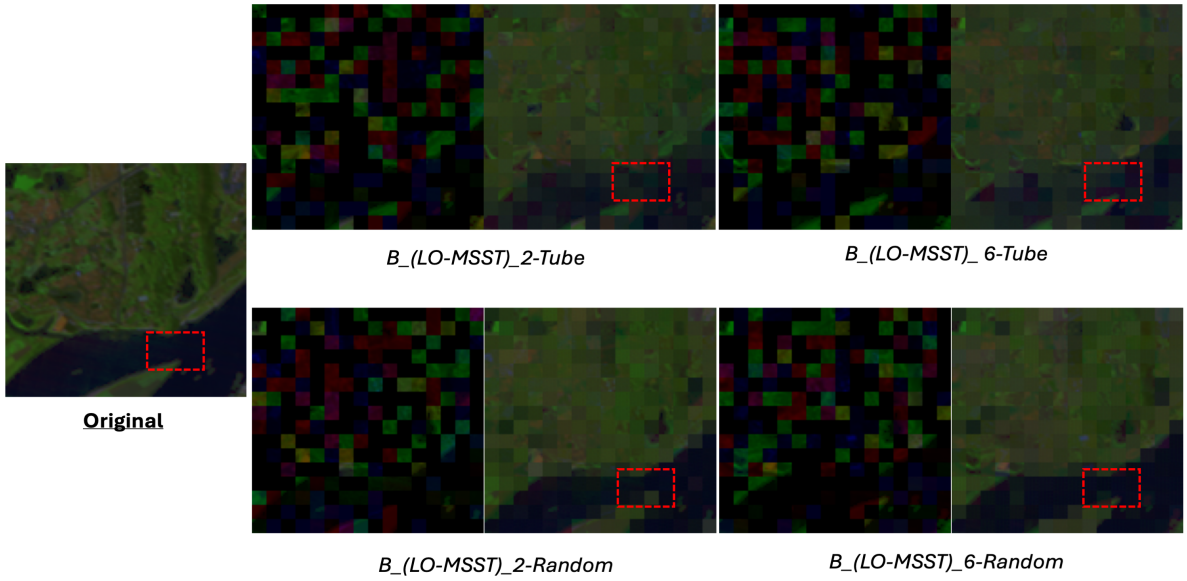

Figure 5. Comparison between the original hyperspectral image and reconstructed images with tube masking and random masking under two pruning configurations for RBG channels [90, 60, 15]. Results show that the base model with random masking consistently outperforms the base model with tube masking in terms of reconstruction quality.

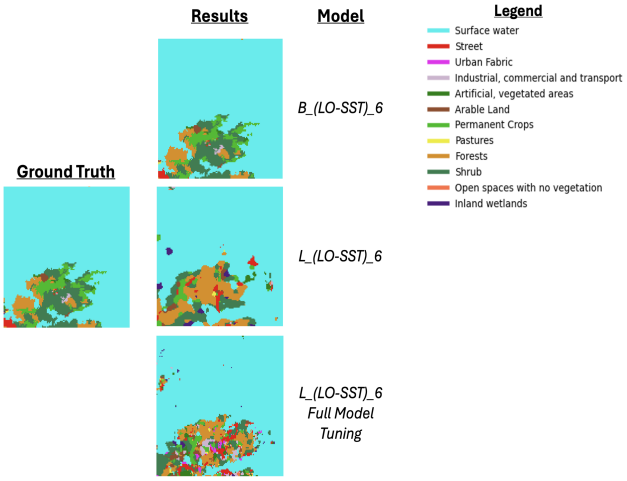

Figure 6. Results for the best-performing model B_(LO-SST)_6 compared to L_(LO-SST)_6 and full model tuned L_(LO-SST)_6 _full. The results show that the base model significantly outperforms both large models.

53.75% due to its multiscale feature aggregation. Interestingly, the base model outperforms the large models across all heads, likely due to better parameter efficiency and handling of hyperspectral data.

Given its simplicity and consistently strong performance, the CNN-Decoder was selected for full model fine-tuning in the subsequent stages and became the basis for all remaining experiments, as it demonstrated effective results for both the large and base configurations.

Table 5. Evaluation of Segmentation Heads with Random and Tubing Masking Pre-training. Pixel Accuracy is reported for different configurations.

Pre-training	Model	Head	Pixel Acc. (%)
Random	B_(LO-SST)_2	UNet	53.45
Random	B_(LO-SST)_4	CNN-Decoder	**55.55**
Random	B_(LO-SST)_4	UNet	54.61
Random	B_(LO-SST)_6	CNN-Decoder	55.16
Random	B_(LO-SST)_6	UNet	55.12
Tubing	B_(LO-SST)_2	CNN-Decoder	48.17
Tubing	B_(LO-SST)_2	UNet	49.64
Tubing	B_(LO-SST)_4	CNN-Decoder	47.83
Tubing	B_(LO-SST)_4	UNet	49.25
Tubing	B_(LO-SST)_6	CNN-Decoder	49.35
Tubing	B_(LO-SST)_6	UNet	51.56

6.2.2 Random vs. Tubing Masking for Pre-Training

We evaluated the performance of segmentation heads under two pre-training strategies: **random masking pre-training** and **tube masking pre-training**. As highlighted in Table 5, random masking pre-training consistently outperformed tube masking pre-training across all configurations. For instance, B_(LO-SST)_4 with a CNN-Decoder head achieved 55.55% accuracy using random masking pre-training, significantly higher than 47.83% with tube masking pre-training. Similarly, B_(LO-SST)_6 with a UNet reached 55.12% accuracy with random pre-training, compared to 51.56% with tube masking pre-training. These findings underscore the effectiveness of **random pre-training** in cap-

Table 6. Segmentation Results for Fine-Tuned Pre-Trained Models and Fine-Tuned Full Models with Base SST: Varying Number of Pruned Layers.

Model	Head	Pre-Trained Acc. (%)	Full Model Acc. (%)
B_(LO-SST)_2	CNN-Decoder	54.89%	51.66%
B_(LO-SST)_4	CNN-Decoder	**55.55%**	51.39%
B_(LO-SST)_6	CNN-Decoder	55.16%	51.43%
L_(LO-SST)_2	CNN-Decoder	41.04%	39.13%
L_(LO-SST)_4	CNN-Decoder	44.47%	39.43%
L_(LO-SST)_6	CNN-Decoder	41.52%	40.02%

turing critical spatial and spectral dependencies for segmentation tasks. Furthermore, these segmentation results align with our observations from the image reconstruction experiments shown in Figure 5, where random masking pre-training demonstrated superior performance in retaining structural and spectral details.

6.2.3 Fine-Tuning the Full Model

In the fine-tuning the full model experimentation phase, the LO-MSST encoder is unfrozen, allowing all parameters to be trained alongside the segmentation head. As shown in Table 6, full model tuning yields mixed results. For instance, the pixel accuracy of B_(LO-SST)_4 drops from 55.55% to 51.39%, while L_(LO-SST)_6 drops from 41.52% to 40.02%, indicating a loss of important pre-trained information in the encoder, such as weights and feature representations. Some models, such as L_(LO-SST)_2, show relatively stable performance after full model tuning, but these cases are less common.

Based on these findings, fully tuning a LO-MSST pre-trained model is suboptimal for downstream tasks such as segmentation. This approach disrupts the features learned during pre-training, particularly when substantial modifications are made to the encoder. Therefore, it is best to avoid full model fine-tuning and instead retain the pre-trained weights, exclusively fine-tuning the segmentation head.

As illustrated in Figure 6, there is a noticeable performance degradation when transitioning from the base configuration to the large configuration, and this effect is further exacerbated during full model fine-tuning.

6.2.4 Model Comparison

To contextualize the performance and efficiency of various models, we present a comparative analysis of their accuracy and parameter count. As shown in Table 7, B-(LO-SST)-4 achieves improved performance with reduced parameters and also exhibits competitive performance compared to much larger models.

Table 7. Model Performance Results. Accuracy and Parameters for Different Models.

Model	Accuracy	Parameters
Mask2Former	53.01%	68,896,057
Base SST	46.51%	8,716,422
Large SST	35.51%	8,988,679
B_(LO-SST)_4	**55.55%**	**8,199,518**
B_(LO-SST)_6	55.16%	8,063,390
L_(LO-SST)_4	44.47%	8,716,423
L_(LO-SST)_6	41.52%	8,580,295

7. Conclusion

This paper introduces the Layer-Optimized Spatial-Spectral Transformer (LO-SST), a model that integrates spatial and spectral attention with structured layer pruning for hyperspectral image segmentation. By pruning less important layers, LO-SST reduces computational complexity while maintaining competitive accuracy, making it an effective solution for resource-constrained hyperspectral data analysis.

Self-supervised pretraining with LO-MSST enables the encoder to learn robust spatial and spectral features. Random masking outperforms tube masking by better preserving critical structural and semantic details, enhancing both reconstruction quality and segmentation performance.

LO-SST pre-training with fine-tuning of the segmentation head outperforms full model tuning, as it retains important weights and feature representations learned during pre-training, which are otherwise lost in full model tuning. These results highlight the effectiveness of efficient pre-training with layer pruning.

In conclusion, these findings position LO-SST as a scalable and efficient approach for advancing hyperspectral data analysis, particularly as transformer networks and large-scale unlabeled datasets gain prominence in remote sensing.

References

[1] Joaquin Alvarez. Confident magnitude-based neural network pruning, 2024. 2

[2] José M Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser M Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2):6–36, 2013. 1, 2

[3] Sarah Chen et al. Evaluating neural network pruning techniques on visual transformers. *CS231n Report*, 2022. 2

[4] Yu Chen, Zhen Guo, and Xiaojie Wang. Hyperspectral image classification with spectral-spatial transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022. 2

[5] Bowen Cheng, Chen Geng, Yuliang Zhu, Mengyuan Li, Lu Jiang, Yanghao Wei, Varun Jampani, Qihang Huang, Ping Luo, Yu-Wing Tai, et al. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 5

[6] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations, 2024. 2

[7] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. In *Advances in Neural Information Processing Systems*, 2022. 2

[8] 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, 2021. 2

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CVPR*, pages 13020–13029, 2020. 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[11] Danfeng Hong, Bing Zhang, Hao Li, Yuxuan Li, Jing Yao, Chenyu Li, Martin Werner, Jocelyn Chanussot, Alexander Zipf, and Xiao Xiang Zhu. Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks, 2023. 5

[12] Damian Ibañez, Ruben Fernandez-Beltran, Filiberto Pla, and Naoto Yokoya. Masked auto-encoding spectral–spatial transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. 3

[13] Junjun Jiang, He Sun, Xianming Liu, and Jiayi Ma. Learning spatial-spectral prior for super-resolution of hyperspectral imagery. *IEEE Transactions on Computational Imaging*, 6:1082–1096, 2020. 2

[14] Yang Liu, Jian Chen, Shuhua Jin, Shijie Wei, Jinjin Hai, Xin Qi, Yongli Li, and Bin Yan. Tube masking-based mae pre-training for three-dimensional lumbar vertebrae segmentation. In *2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 1161–1164, 2024. 3

[15] Pengyuan Lv, Wenjun Wu, Yanfei Zhong, and Liangpei Zhang. Review of vision transformer models for remote sensing image scene classification. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 2231–2234, 2022. 2

[16] Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. A survey on efficient vision transformers: Algorithms, techniques, and performance benchmarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7682–7700, Dec. 2024. 2

[17] Rakiba Rayhana, Zhenyu Ma, Zheng Liu, Gaozhi Xiao, Yuefeng Ruan, and Jatinder S. Sangha. A review on plant disease detection using hyperspectral imaging. *IEEE Transactions on AgriFood Electronics*, 1(2):108–134, 2023. 2

[18] Linus Scheibenreif, Michael Mommert, and Damian Borth. Masked vision transformers for hyperspectral image classification. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2166–2176, 2023. 2, 3

[19] Zhenwei Shi and Zhibin Yang. Hyperspectral image classification and dimensionality reduction: An overview. *Journal of Sensors*, 2019. 1, 2

[20] Xinya Wang, Qian Hu, Yingsong Cheng, and Jiayi Ma. Hyperspectral image super-resolution meets deep learning: A survey and perspective. *IEEE/CAA Journal of Automatica Sinica*, 10(8):1668–1691, 2023. 2

[21] Zhirong Xu, Chen Wei, et al. Masked autoencoders are robust data augmentors. *arXiv preprint arXiv:2206.04846*, 2022. 2

[22] Huanrui Yang et al. Global vision transformer pruning with hessian-aware saliency. *CVPR*, 2023. 2

[23] Lu Yu et al. X-pruner: explainable pruning for vision transformers. *CVPR*, 2023. 2

[24] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. A survey on masked autoencoder for self-supervised learning in vision and beyond, 2022. 2

[25] Yuan Zheng, Jian Wang, and Jianjun Tang. Spatial-spectral transformers for hyperspectral image classification. *Journal of Remote Sensing*, 12(3):553–567, 2021. 2