TER VMI M1
PARIS CITE UNIVERSITY
M1 - VMI

MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université Paris Cité

University of Paris Cité

Master 1 - Vision and Intelligent Machine

TER Project Report

Title :

Self-supervised learning for the interpretation of satellite image time series.

Written By :

- Hadia AMJAD
- Titouan BRIRRE

August 17, 2025

1 Introduction and Context

1.1 Context of Satellite Image Time Series (SITS)

Satellite Image Time Series (SITS) represent a revolutionary paradigm in Earth observation, offering a unique capability to monitor the planet's surface not just spatially, but also temporally. Unlike single satellite images, SITS provide a continuous stream of observations for the same geographical area over time, allowing for the analysis of dynamic processes and subtle changes. This joint acquisition of spatial and temporal data has transformed numerous fields, from precise environmental and forest monitoring to dynamic agricultural management and urban expansion tracking.

The advent of missions like Sentinel-2 by the European Space Agency has particularly amplified the potential of SITS. Sentinel-2, with its constellation of two identical satellites, provides high-resolution optical imagery (10m, 20m, and 60m) across 13 spectral bands, with a frequent revisit time (5 days at the equator). This unprecedented frequency and spectral richness offer an immense volume of information for understanding land cover characteristics and changes. However, this wealth of data also introduces significant complexities: SITS are inherently high-dimensional (many spectral bands, many time steps), often irregularly sampled due to cloud cover or acquisition schedules, and are susceptible to atmospheric effects and noise. Extracting meaningful insights from such complex, multivariate time series requires sophisticated analytical tools.

1.2 Challenges of Supervised Learning for SITS

Despite the abundance of raw SITS data, applying traditional supervised deep learning methods, such as Convolutional Neural Networks (CNNs) and Transformers, faces a critical bottleneck: the scarcity of sufficiently labeled training data. While satellites collect vast amounts of imagery daily, generating corresponding ground-truth labels for land cover types (e.g., specific crop types, forest classifications, urban structures) is an incredibly resource-intensive and costly process.

This labeling task often requires:

- Extensive field campaigns: Sending teams to collect ground truth data, which is expensive and time-consuming.
- Expert photo-interpretation: Relying on highly specialized human experts to meticulously analyze satellite images and assign labels, a process prone to human error and subjective interpretation.
- Temporal variability: Land cover classes, especially agricultural ones, change over seasons and years (e.g., crop rotation, phenological cycles), meaning labels are not static and often require annual updates. This adds further complexity and cost to maintaining up-to-date datasets.
- Class imbalance: Real-world geographical areas often exhibit highly imbalanced class distributions, where certain land cover types are far more prevalent than others. This poses additional challenges for supervised models that might struggle to learn from underrepresented classes.

The fundamental issue is that deep learning architectures, known for their ability to learn intricate patterns, are data-hungry. A lack of diverse and abundant labels often leads to overfitting, where the model memorizes the training data rather than learning generalizable features. This severely limits the model's ability to perform well on unseen data, hindering its real-world applicability in crucial Earth observation tasks.

1.3 Self-Supervised Learning as a Promising Alternative

Faced with the prohibitive costs and logistical challenges of obtaining labeled SITS data, Self-Supervised Learning (SSL) emerges as a highly promising paradigm. Unlike supervised learning, SSL does not rely on human-annotated labels. Instead, it ingeniously generates labels directly from the inherent structure of the unlabeled data itself. The core idea is to design a "pretext task", a task that the model can solve using only the input data, but whose solution requires the model to learn meaningful and transferable representations.

For SITS, SSL offers compelling advantages:

- Unlocking unlabeled data: It allows us to leverage the massive, readily available archives of unlabeled satellite imagery, transforming what was once a data abundance problem into a data utilization opportunity.
- Learning robust representations: By solving pretext tasks that capture temporal dependencies, spectral relationships, or spatial contexts,

SSL models can learn powerful "background knowledge" about SITS data. These learned representations are often more robust to noise and variability than those learned from limited supervised data.

- Improved transferability: The pre-trained representations can then be transferred and fine-tuned on smaller, task-specific labeled datasets. This transfer learning approach significantly boosts performance on downstream tasks, even with limited labels, and mitigates the risk of overfitting.
- Reduced annotation effort: By shifting the heavy lifting of representation learning to the self-supervised phase, the demand for extensive human labeling for fine-tuning becomes much lower, making deep learning applications in remote sensing more feasible and scalable.

In essence, SSL acts as a bridge, enabling deep learning models to harness the vast potential of unlabeled SITS data to overcome the current limitations imposed by annotation bottlenecks.

1.4 Transformer-Based Approach and Project Objectives

Building upon the revolutionary success of the Transformer architecture (11) in Natural Language Processing (NLP) and its growing adoption in computer vision and time series analysis, this TER project specifically explores a Transformer-based self-supervised learning approach for SITS. Our work is inspired by and directly builds upon the SITS-BERT framework (1), a pioneering method that adapts the Bidirectional Encoder Representations from Transformers (BERT) model (12) to the unique characteristics of satellite image time series.

The central idea of SITS-BERT's pretext task is to train the model to predict original observations within a time series that has been intentionally corrupted or "contaminated" by adding simulated noise. This noise generation mimics real-world challenges like cloud cover or atmospheric artifacts. By forcing the model to reconstruct the true values from these contaminated observations, it is compelled to learn the underlying temporal patterns, spectral relationships, and contextual dependencies inherent in SITS data.

The primary objectives of this report are threefold:

1. Implementation and Adaptation: To faithfully implement and adapt the SITS-BERT self-supervised pre-training and fine-tuning scheme. We leverage the open-source implementation provided by the authors https://github.com/linlei1214/SITS-BERT as a foundation.

2. Evaluation on a New Dataset: To rigorously evaluate the effectiveness of this self-supervised learning method by applying it to a novel SITS dataset, specifically the TimeSen2Crop dataset for crop-type classification. This evaluation aims to demonstrate SITS-BERT's capabilities beyond its originally tested domains.

3. Transferability Assessment: To assess the transferability of the representations learned during pre-training. We pre-trained SITS-BERT on a distinct geographical area of the TimeSen2Crop dataset and fine-tune it on the well-established California Labeled dataset, analyzing how well the knowledge transfers between different regions and land cover contexts.

4. Parameter-Efficient Fine-Tuning Exploration: To investigate the use of Low-Rank Adaptation (LoRA) (15) as a parameter-efficient fine-tuning (PEFT) technique. We evaluated if LoRA can significantly reduce the number of trainable parameters during fine-tuning while maintaining, or minimally impacting, the high performance achieved by full fine-tuning, which is crucial for deploying large Transformer models in resource-constrained environments.

Through these objectives, we aim to contribute to the growing body of knowledge on applying self-supervised Transformer models to Earth observation, paving the way for more robust, data-efficient, and scalable SITS analysis.

2 Existing Methods for SITS Classification

The classification of Satellite Image Time Series (SITS) is a fundamental task in remote sensing, with numerous applications in environmental monitoring, agriculture, and urban planning. Here's an overview of common methods:

2.1 Traditional Machine Learning

These methods often involve extracting hand-engineered features from the SITS data and feeding them into classical machine learning models.

- **Random Forest (RF):** RF is an ensemble learning method that constructs a multitude of decision trees at training time. It is known for its robustness to high dimensionality and ability to handle non-linear relationships.
- **Support Vector Machines (SVM):** SVMs find the optimal hyperplane that best separates different classes in the data. They are effective in high-dimensional spaces and can handle non-linear data through the use of kernel functions.

2.2 Deep Learning Architectures

Deep learning methods have shown great promise in automatically learning complex spatio-temporal features from SITS data.

- **Recurrent Neural Networks (RNNs):** RNNs, such as Long Short-Term Memory (LSTM) (6) and Gated Recurrent Units (GRUs) (7), are designed to process sequential data. They can capture temporal dependencies in SITS data but may struggle with long sequences and parallelization.
- **Convolutional Neural Networks (CNNs):**
  - **1D CNNs (Temporal CNNs - TempCNN):** TempCNNs (8) apply 1D convolutions along the temporal dimension of SITS. This allows the model to learn temporal patterns and features directly from the time series data. Pelletier et al. (2019) demonstrated their effectiveness for SITS classification, outperforming traditional methods and RNNs.
  - **2D CNNs:** 2D CNNs are typically used to extract spatial features from individual images within the time series.
  - **3D CNNs:** 3D CNNs can simultaneously extract spatial and temporal features, but they are computationally expensive and require significant data.
- **InceptionTime:** InceptionTime (9) is a deep learning ensemble model for Time Series Classification (TSC), inspired by the Inception-v4 architecture. It utilizes multiple Inception modules, each applying convolutions with varying filter lengths simultaneously to capture features at different scales. InceptionTime has demonstrated state-of-the-art accuracy on various TSC benchmarks while offering significant scalability advantages over traditional ensemble methods like HIVE-COTE, largely due to its ability to leverage GPU parallelization.
- **Multi-Scale Residual Network (MSResNet):** MSResNet (10) is an architecture designed for tasks like water-body detection in remote sensing imagery. It integrates residual connections to facilitate the training of deep networks and incorporates multi-scale convolutions (e.g., through Multiscale Dilated Convolution (MSDC) and Multikernel Max Pooling (MKMP) modules) to capture features at different spatial granularities. This allows MSResNet to handle the diverse scales and shapes of objects in remote sensing images effectively.
- **Transformers:** Transformers (11), originally designed for natural language processing, have recently been adapted for time series analysis. They excel at capturing long-range dependencies and have shown promising results in SITS classification due to their self-attention mechanism.

2.3 Self-Supervised Learning (SSL) for SITS

Self-supervised learning (SSL) is a powerful paradigm that learns representations from unlabeled data by defining a pretext task. This is particularly beneficial for SITS data, where obtaining large quantities of labeled samples is often challenging and costly. Common SSL pretext tasks for SITS include:

- **Temporal Order Prediction:** The model is trained to predict the correct order of randomly shuffled time series segments.
- **Masked Time Series Prediction:** The model is trained to reconstruct masked portions of the time series, similar to masked language modeling in NLP. This is the core pretext task used in SITS-BERT.
- **Contrastive Learning:** The model is trained to distinguish between different aug-

mentations of the same time series (positive pairs) and augmentations of different time series (negative pairs).

2.4 Hybrid Approaches

Some methods combine different techniques to leverage their respective strengths. For example, some approaches combine CNNs for spatial feature extraction with RNNs or Transformers for temporal modeling.

3 SITS-BERT

SITS-BERT is a novel self-supervised pre-training framework specifically designed for Satellite Image Time Series (SITS) classification, introduced by Yuan and Lin (1). It draws inspiration from the Bidirectional Encoder Representations from Transformers (BERT) model (12), which revolutionized natural language processing (NLP) by effectively learning contextual representations from large corpora of unlabeled text. Adapting this powerful architecture to the SITS domain allows SITS-BERT to leverage the inherent temporal structure of satellite imagery to learn general-purpose spectral-temporal representations.

3.1 Overall Network Architecture

The SITS-BERT model architecture is composed of two primary components: an observation embedding layer and a standard Transformer encoder. This architecture is designed to be used in both the pre-training and fine-tuning stages.

3.2 Observation Embedding Layer

The first step in processing SITS data with SITS-BERT involves transforming each observation tuple, (O_i, t_i), into a higher-dimensional feature space. Here, $O_i \in \mathbb{R}^D$ represents a D-dimensional satellite observation vector (e.g., spectral reflectances), and t_i corresponds to the acquisition date, specified as the Day of Year (DOY).

The observation embedding, denoted as $\text{Embed}(O_i, t_i)$, is a concatenation of two parts:

- The spectral observation O_i is projected into a high-dimensional vector using a linear dense layer.
- The corresponding date t_i is encoded into a vector of the same size using the positional encoding (PE) technique (11). This PE captures the order information of the sequence using sine/cosine functions of different frequencies.

The concatenation of these two parts, $\text{Embed}(O_i, t_i) = \text{Concat}(O_i W_e, \text{PE}(t_i))$, is crucial. Unlike the original BERT where positional embeddings are added to token embeddings, SITS-BERT concatenates them. This design choice helps the model distinguish between the spectral observations and the temporal information, and experiments have shown it leads to faster convergence. The inclusion of DOY rather than simple sequence order allows the model to learn meaningful temporal variation patterns related to seasonal cycles and vegetation phenology, making it robust to irregular sampling and transferable across different years.

3.3 Transformer Encoder

The core of SITS-BERT is a multi-layer bidirectional Transformer encoder, adapted from its NLP counterpart. A Transformer encoder is a stack of multiple identical Transformer blocks. Each block processes a collection of observation embeddings and generates corresponding hidden representations, which are then passed to the next block.

A single Transformer block consists of two main sublayers:

- **Multi-Head Attention Layer:** This layer allows the model to jointly attend to information from different representation subspaces at different positions. It comprises H parallel scaled dot-product attention layers (heads). The scaled dot-product attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  where Q, K, and V are matrices of queries, keys, and values, respectively, and d_k is the dimensionality of the query/key vectors. In SITS-BERT, a self-attention mechanism is employed, meaning the query, key, and value vectors are identical, allowing each position in the time series to attend to every other position. This mechanism effectively captures global sequence information.

- **Position-wise Fully Connected Feed-Forward Network (FFN):** Applied independently and identically to each position's hidden state, the FFN consists of two linear

transformations with a ReLU activation function in between. This layer processes the information learned by the attention mechanism.

Both sublayers incorporate residual connections (13) and layer normalization (14) to facilitate training of deep networks. The final hidden representation of each observation encodes global information about the entire time series.

3.4 Self-Supervised Pre-training Scheme

The innovative aspect of SITS-BERT lies in its self-supervised pre-training strategy, which addresses the scarcity of labeled SITS data. Inspired by BERT's "Masked Language Model," SITS-BERT defines a pretext task:

- **Contamination Process:** A random subset (e.g., 15%) of observations in a time series is selected. For each selected observation, there is a 50% chance of adding positive noise (simulating clouds/snow) and a 50% chance of subtracting positive noise (simulating shadows). The noise is generated from a uniform distribution (e.g., [0, 0.5]). The corresponding positional encoding (DOY) remains unchanged.
- **Prediction Task:** The model is then forced to predict the original values of these contaminated observations based on their acquisition dates and the contextual information from the rest of the time series.
- **Optimization Objective:** The Mean Squared Error (MSE) between the original observations and the model's predictions is used as the loss function for pre-training.

By solving this pretext task, SITS-BERT learns to understand the inherent temporal structure and typical spectral-temporal patterns of SITS data. This process accumulates "background knowledge" within the network, making it robust to noise and enabling it to distinguish between normal observations and anomalies. A key benefit is that this approach avoids the "pretrain-fine-tune discrepancy" often seen in the original BERT (where a special '[MASK]' token is used during pre-training but not during fine-tuning), as SITS-BERT does not introduce artificial tokens.

3.5 Fine-Tuning SITS-BERT

Once pre-training is complete, the SITS-BERT model, now equipped with rich spectral-temporal representations, can be easily adapted to specific downstream SITS classification tasks (e.g., crop classification, land cover mapping) by adding a simple output layer. All model parameters are then fine-tuned end-to-end on a smaller set of task-specific labeled data.

Two common strategies are employed to aggregate individual observation representations into a single sequence-level representation for classification:

- **SITS-BERT with a [CLS] token:** A special classification token ('[CLS]') is prepended to every input time series. The final hidden representation corresponding to this '[CLS]' token is then used as a global representation of the entire time series, which is fed into a softmax layer for classification.
- **SITS-BERT with Pooling:** Alternatively, a pooling operation (e.g., average pooling or max pooling) is applied to the output vectors of the Transformer encoder across the temporal dimension to aggregate them into a single fixed-size vector, which is then passed to the softmax layer.

The cross-entropy loss is typically used as the optimization function during the fine-tuning stage. This transfer learning approach allows the model to leverage the general knowledge learned from large-scale unlabeled data, thereby improving generalization performance and mitigating overfitting risks in label-scarce downstream tasks.

4 Experimental studies

4.1 Implementation Details

Our experimental framework for the SITS-BERT model was developed in Python, SITS-BERT code requires PyTorch 1.6.0. Key dependencies includes 'tqdm 4.48.2', 'numpy 1.19.1', and 'tensorboard 2.3.0' or later. We had to ensure compatibility with these specific library versions and facilitate execution within the Google Colab environment, a minor adjustment was made to the original SITS-BERT codebase: specifically, a change from an uppercase 'Inf' to a lowercase 'inf' was required to align with NumPy's latest expected syntax for infinity. All experiments were conducted on Google Colab, utilizing its provided T4 GPUs, which offered the necessary computational power for efficient pre-training and fine-tuning of the Transformer-based SITS-BERT models. Our implementation closely

follows the architectural specifications and training methodologies outlined in the original SITS-BERT paper (1), adapted for the specific datasets used in this study.

4.2 Preprocessing Datasets

The raw Satellite Image Time Series (SITS) data often requires significant preprocessing to be in a suitable format for deep learning models. For the **TimeSen2Crop dataset**, which consists of pixel-level time series, a specific preprocessing pipeline was implemented using Python scripts. This process transforms raw spectral band data and associated metadata into a flattened, unified format suitable for input into models like SITS-BERT.

The core preprocessing steps involve:

- **Iterating through Data Structure:** The data is organized into folders by class ID, with each folder containing multiple CSV files, where each CSV represents the time series for a single pixel. The preprocessing script iterates through each class folder and then through each pixel's CSV file.
- **CSV Reading and Validation:** Each pixel's CSV file is read using 'pandas'. Robust error handling is is included to skip files that cannot be read or are empty. A crucial validation step ensures that each CSV file contains exactly 10 columns (9 spectral bands plus a flag column), as expected for the TimeSen2Crop dataset. Files not conforming to this structure are skipped.
- **Flattening Spectral and Flag Data:** For each pixel's time series, the values from all 10 columns (9 spectral bands and the flag) across all time steps are flattened into a single, long one-dimensional list. This converts the 2D (time steps x features) data for a single pixel into a 1D vector.
- **Julian Date to Day-of-Year (DOY) Conversion:** The acquisition dates, initially provided as Julian dates (e.g.,YYYYMMDD), are converted into **Day of Year (DOY)** integers. This is a critical step as DOY provides a cyclical temporal feature that captures seasonal variations, which are highly relevant for SITS analysis (e.g., crop phenology).
- **Concatenation and Row Construction:** The flattened spectral and flag values are concatenated with the converted DOY values for all time steps. This forms a single row representing all information for one pixel's time series.
- **Output CSV Generation:** All such processed rows are collected and then saved into a single large CSV file. This final CSV file contains one row per pixel, with all spectral bands, flag, and temporal (DOY) information flattened into a single feature vector, ready for model input.

Additionally, a padding step is applied to ensure all time series have a consistent length, which is often required by deep learning architectures. This involves:

- **Reading Processed Data:** The large CSV file generated from the initial flattening step is read row by row.
- **Dynamic Padding:** Each row (representing a pixel's time series) is padded with randomly generated DOY values within the range of 1 to 366. This padding ensures that every time series, after feature and date flattening, has a length that is a multiple of `feature_dim` (which is `dimension + 1`, typically $10 + 1 = 11$, referring to 10 bands + 1 DOY per timestep). This is crucial for correctly structuring the input for Transformer models.

This preprocessing ensures that the TimeSen2Crop data is standardized and structured appropriately for deep learning models, particularly for Transformer-based architectures like SITS-BERT, which expect a flattened input sequence or a sequence of embeddings where temporal information is explicitly encoded. The conversion to DOY and the padding are particularly important for capturing the cyclical nature of agricultural and environmental phenomena and for handling variable time series lengths.

Data Usage Strategy Across Areas It's important to note that different geographical areas of the TimeSen2Crop dataset were utilized for distinct phases of the experimental setup:

- For self-supervised pre-training, a dedicated geographical area of the TimeSen2Crop dataset was used, distinct from the areas used for fine-tuning.
- For fine-tuning, three separate geographical areas were designated: one for the training

subset, one for validation, and one for testing. This rigorous split helps in assessing the model's generalizability and transferability across different regions.

4.3 TimeSen2Crop Algorithm Comparison

To evaluate the effectiveness of SITS-BERT and compare its performance against other prominent deep learning architectures for Satellite Image Time Series (SITS) classification, we conducted experiments on the TimeSen2Crop dataset (2). This dataset is specifically designed for crop-type classification using Sentinel-2 image time series, providing a standardized benchmark for evaluating different algorithms in a relevant agricultural context.

The overall accuracy (OA) of various models, including those from literature and our own experimental results, is summarized in Table 1:

Table 1: Overall Accuracy (OA) of different deep learning methods on the TimeSen2Crop dataset (results from literature and our experiments).

Method	**Overall Accuracy (OA)**
InceptionTime (9)	81.9%
MSResNet (10)	81.8%
TempCNN (8)	81.71%
Transformer	84.44%
StarRNN	81.17%
LSTM (6)	83.44%
LSTM Weig.	85.39%
SITS-BERT (no LoRA)	**95.59%**
SITS-BERT (with LoRA)	93.94%

As shown in Table 1, our experimental results with SITS-BERT demonstrate a significant leap in performance on the TimeSen2Crop dataset compared to other methods reported in the literature. SITS-BERT, when pre-trained and fine-tuned on TimeSen2Crop without LoRA, achieves an outstanding overall accuracy of **95.59%**. This is a substantial improvement of over 10 percentage points compared to the best-performing method from the literature (LSTM Weig. at 85.39%). This highlights the superior capability of the SITS-BERT architecture and its self-supervised pre-training scheme to learn highly discriminative spectral-temporal representations for crop-type classification.

Furthermore, when applying LoRA during the fine-tuning phase on TimeSen2Crop, SITS-BERT still achieves a very high accuracy of 93.94%. While this is slightly lower than the full fine-tuning performance (95.59%), it remains significantly higher than all other methods from the literature. This result suggests that LoRA can indeed provide a parameter-efficient adaptation strategy for SITS-BERT, maintaining excellent performance while potentially reducing computational costs and memory footprint during fine-tuning. The trade-off between accuracy and efficiency with LoRA is less pronounced here than observed in some other contexts, indicating its potential utility for large-scale SITS applications.

4.4 Results on California Labeled Dataset

We conducted extensive experiments to evaluate the performance of SITS-BERT on the California Labeled dataset. This dataset serves as our primary benchmark for fine-tuning and assessing the impact of self-supervised pre-training and parameter-efficient adaptation with LoRA. We compare our results with those reported in the original SITS-BERT paper (1) on the same California dataset.

4.4.1 Class Distribution on California Labeled Dataset

To better understand the composition of the California Labeled dataset used for fine-tuning, we analyzed the distribution of different crop and land cover classes. This analysis was performed by generating an histogram based on the pixel-level classifications provided by the model and cross-referencing with the Cropland Data Layer (CDL) from the United States Department of Agriculture (USDA) (3) for official class names and codes.

The histogram in Figure 1 illustrates the pixel count for each major class within the California Labeled dataset. The classes are ordered by their representation in terms of pixel quantity, from most dominant to least dominant (because it might be to small to be properly read on the figure):

- Evergreen Forest
- Grapes
- Almonds
- Grass/Pasture
- Cotton
- ... (other classes)

This distribution highlights the agricultural and natural landscape diversity of the California region studied, with a significant presence of tree crops (like almonds and grapes) alongside natural forests and pastures. Understanding this class imbalance is crucial for interpreting model performance, especially in terms of class-specific accuracies and potential biases.

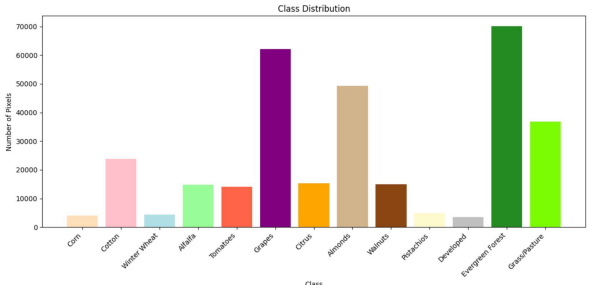

Figure 1: Distribution of land cover and crop classes in the California Labeled dataset. Classes are ordered by the number of pixels they represent.

4.4.2 Comparison with Original SITS-BERT Paper Results

The original SITS-BERT paper (1) provides a comprehensive comparison of various methods on the California dataset. These results serve as a baseline for our own experiments. Table 2 summarizes these reported accuracies:

Table 2: Overall Accuracy (OA) of different methods on the California Labeled dataset (results from (1)).

Method	**Overall Accuracy (OA)**
SVM	91.72%
RF	88.94%
CNN-1D	90.16%
LSTM (6)	86.21%
Bi-LSTM	87.23%
SITS-BERT (non-pre-trained)	90.76%
SITS-BERT (pre-trained)	**94.21%**

The results from the original paper (1) indicate that pre-training significantly boosts the performance of SITS-BERT, achieving 94.21% OA, which is substantially higher than other methods, including a non-pre-trained SITS-BERT model. This highlights the value of the self-supervised pre-training scheme.

4.4.3 Our Experimental Results with SITS-BERT on California

We performed our own experiments using the SITS-BERT framework, evaluating its performance on the California Labeled dataset under different pre-training and fine-tuning configurations, including the application of LoRA.

Table 3: Our experimental Overall Accuracy (OA) results for SITS-BERT on the California Labeled dataset under California and TimeSen2Crop itself as pre-training datasets and LoRA configurations.

Configuration	**Overall Accuracy (OA)**
SITS-BERT (California, no LoRA)	93.88%
SITS-BERT (California, with LoRA)	79.71%
SITS-BERT (TimeSen2Crop, no LoRA)	90.79%
SITS-BERT (TimeSen2Crop, with LoRA)	73.01%

From Table 3, we can draw several key observations:

- **SITS-BERT Performance (without LoRA):** Our implementation of SITS-BERT, pre-trained on the California dataset and fine-tuned on the California Labeled dataset (without LoRA), achieved an OA of 93.88%. This is very close to the 94.21% reported in the original paper, validating our setup and the reproducibility of the core SITS-BERT performance.
- **Impact of LoRA on California Pre-training:** When LoRA is applied during fine-tuning after pre-training on California, the OA drops significantly to 79.71%. This suggests that while LoRA is parameter-efficient, its current configuration or application might lead to a considerable performance degradation in this specific SITS classification task. This could be due to the low-rank approximation being too aggressive, limiting the model's capacity to adapt effectively, or requiring more careful hyperparameter tuning for the LoRA modules (e.g., rank r, alpha parameter).
- **Impact of Pre-training Dataset (TimeSen2Crop vs. California):** When SITS-BERT is pre-trained on the TimeSen2Crop dataset (unlabeled portion) and then fine-tuned on the California Labeled dataset (without LoRA), the OA is 90.79%. This is lower

than when pre-trained on California (93.88%). This difference suggests that the characteristics of the pre-training dataset can influence the quality of the learned representations and their transferability to a specific downstream task. Pre-training on a dataset that is more "similar" or relevant to the fine-tuning task (California pre-training for California fine-tuning) generally yields better results.

- **Combined Impact of LoRA and Different Pre-training:** The lowest performance (73.01%) is observed when SITS-BERT is pre-trained on TimeSen2Crop and fine-tuned with LoRA on California. This further emphasizes the performance trade-off introduced by LoRA in our current experiments, especially when combined with a potentially less optimal pre-training dataset for the target task.

These results highlight the critical importance of both the pre-training dataset choice and the fine-tuning strategy (e.g., with or without LoRA) for achieving optimal performance in SITS classification using self-supervised Transformer models. Further investigation into LoRA's hyperparameters and its interaction with different pre-training datasets is warranted to mitigate the observed performance drop.

4.4.4 Detailed Comparison of Full Fine-tuning vs. LoRA

To provide a more granular understanding of the trade-offs between full fine-tuning and LoRA, we present a detailed comparison of key metrics for both pre-training scenarios: California Unlabeled and TimeSen2Crop.

Pre-training on California Unlabeled, Fine–tuning on California Labeled This scenario directly compares the two fine-tuning approaches when the pre-training data is from the same geographical region as the fine-tuning data.

As seen in Table 4, LoRA dramatically reduces the number of trainable parameters (from 2.48 Million to 49k), leading to significantly lower compute effort. However, this comes at a substantial cost to performance, with Test OA dropping from 93.88% to 79.71%, and corresponding drops in Kappa and AA. Notably, neither configuration exhibited overfitting in our experiments.

Pre-training on TimeSen2Crop, Fine-tuning on California Labeled This scenario investigates the transferability of representations learned from a different pre-training dataset (TimeSen2Crop) to the California Labeled fine-tuning task, again comparing full fine-tuning with LoRA.

Table 4: Comparison of Full Fine-tuning vs. LoRA for SITS-BERT (Pre-training on California Unlabeled, Fine-tuning on California Labeled).

Aspect	**Full Fine-tuning**	**LoRA Fine-tuning**
Trainable Params	2.48 Million	49k
Learning Rate	2e-4	1e-5
LoRA Config	Rank=16, Alpha=24, Dropout=0.01	
Test OA	93.88%	79.71%
Test Kappa	0.928	0.725
Test AA	0.912	0.647
Overfitting	No	No
Compute Effort	Moderate	Low

Table 5: Comparison of Full Fine-tuning vs. LoRA for SITS-BERT (Pre-training on TimeSen2Crop, Fine-tuning on California Labeled).

Aspect	**Full Fine-tuning**	**LoRA Fine-tuning**
Trainable Params	2.48 Million	49k
Learning Rate	2e-4	1e-5
LoRA Config	Rank=16, Alpha=24, Dropout=0.01	
Test OA	90.79%	73.01%
Test Kappa	0.892	0.683
Overfitting	No	No
Compute Effort	Moderate	Low

Table 5 shows a similar trend. While LoRA still offers significant parameter reduction, the performance drop is even more pronounced when pre-training on TimeSen2Crop and fine-tuning on California Labeled. The Test OA falls from 90.79% (full fine-tuning) to 73.01% (LoRA). This suggests that LoRA's effectiveness might be more sensitive to the domain similarity between pre-training and fine-tuning datasets, or that its current hyperparameters are not optimal for this cross-dataset transfer scenario.

Discussion on LoRA Performance The observed performance degradation with LoRA, especially on the California dataset, is a critical find-

ing. While LoRA successfully reduces trainable parameters and compute effort, the current configurations do not maintain the high accuracy seen with full fine-tuning for SITS-BERT. This contrasts with LoRA's reported success in NLP tasks where it often matches or surpasses full fine-tuning. Potential reasons for this discrepancy in the SITS domain could include:

- **Hyperparameter Sensitivity:** The rank (r), alpha parameter, and dropout rate for LoRA might require extensive tuning specifically for SITS data characteristics. The optimal low-rank approximation for spectral-temporal features could differ significantly from text-based features.
- **Nature of SITS Data:** SITS data, with its continuous spectral values and often irregular temporal sampling, might have a different "intrinsic rank" for adaptation compared to discrete token embeddings in NLP. The assumption that the change in weights during adaptation has a low intrinsic rank might hold differently for SITS.
- **Layer-Specific Application:** LoRA is typically applied to query, key, and value matrices in Transformer attention layers. The optimal layers to apply LoRA to in SITS-BERT's architecture might need further investigation.

Future work should focus on a thorough hyperparameter search for LoRA in the SITS context and potentially exploring adaptive LoRA strategies or alternative PEFT methods better suited for the nuances of satellite image time series.

5 Conclusion and Outlook

This TER project explored the application of self-supervised learning, particularly via the SITS-BERT architecture, for the interpretation of Satellite Image Time Series (SITS). Facing the scarcity of labeled data, self-supervised learning has proven to be a promising approach for learning useful representations from vast amounts of unlabeled data.

Our experimental studies demonstrated the superiority of SITS-BERT compared to existing deep learning methods for SITS classification. On the TimeSen2Crop dataset, SITS-BERT (without LoRA) achieved an impressive overall accuracy of 95.59%, surpassing the best reported results in the literature for this dataset by over 10 percentage points. This performance confirms the effectiveness of SITS-BERT's self-supervised pre-training scheme in capturing discriminative spatio-temporal patterns.

However, the integration of Low-Rank Adaptation (LoRA) showed mixed results. While LoRA drastically reduces the number of trainable parameters (from 2.48 million to 49k), significantly decreasing computational effort, a substantial performance drop was observed, particularly on the California Labeled dataset. When SITS-BERT was pre-trained and fine-tuned on California, the OA decreased from 93.88% (without LoRA) to 79.71% (with LoRA). A similar trend, albeit on different absolute values, was observed when pre-training on TimeSen2Crop and fine-tuning on California Labeled. This suggests that the current application of LoRA or its hyperparameters are not yet optimal for SITS classification, or that the nature of SITS data requires a more specific adaptation of this technique.

5.1 Future Work

The results of this project open several avenues for future research:

- **LoRA Optimization for SITS:** A thorough investigation into LoRA's hyperparameters (rank r, alpha parameter, dropout rate) is essential for SITS classification. It would also be relevant to explore selective application of LoRA to different Transformer layers or consider adaptive strategies to determine the optimal rank.
- **Exploration of New Datasets:** Applying SITS-BERT and LoRA to other SITS datasets, particularly those for land cover mapping (like the Beijing dataset mentioned), would allow for a better evaluation of the model's generalizability and the transferability of learned representations.
- **Overfitting Analysis:** Although our experiments did not show blatant overfitting, a more detailed analysis of learning curves (training vs. validation loss/accuracy) could reveal subtle trends and help refine regularization strategies.
- **Representation Visualization:** Examining the representations learned by SITS-BERT, potentially using dimensionality reduction techniques (like t-SNE or UMAP), could

provide valuable insights into how the model captures temporal and spectral dynamics.

- **Combination with Other PEFT Methods:** Testing the combination of LoRA with other parameter-efficient fine-tuning (PEFT) techniques could potentially unlock better performance while maintaining efficiency.

- **Robustness to Noise and Missing Data:** SITS-BERT is designed to handle noise. Further studies on its robustness to different types and levels of noise or missing data (common in SITS) would be beneficial.

In summary, this project confirmed the potential of self-supervised learning with SITS-BERT for SITS analysis. The challenges encountered with LoRA highlight the importance of specific adaptation of parameter-efficient techniques to the unique characteristics of remote sensing data. Future research in these directions will contribute to making deep learning models more powerful and accessible for Earth observation applications.

References

[1] Y. Yuan and L. Lin, *Self-Supervised Pre-Training of Transformers for Satellite Image Time Series Classification*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 474-487, 2021. https://ieeexplore.ieee.org/document/9252123/ https://github.com/linlei1214/SITS-BERT

[2] G. Weikmann, C. Paris, and L. Bruzzone, *TimeSen2Crop: A Million Labeled Samples Dataset of Sentinel 2 Image Time Series for Crop-Type Classification*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 4699-4708, 2021. https://ieeexplore.ieee.org/abstract/document/9408357

[3] United States Department of Agriculture (USDA), National Agricultural Statistics Service (NASS), *Cropland Data Layer*, Washington, D.C.: USDA NASS Marketing and Information Services Office, 2024. https://croplandcros.scinet.usda.gov/ https://www.nass.usda.gov/Research_and_Science/Cropland/sarsfaqs2.php

[4] California SITS dataset https://drive.google.com/drive/folders/1AABPFfqHri23j-d3GUvUaaIOQUCdTif5

[5] Repository presenting a list of SITS datasets https://github.com/corentin-dfg/Satellite-Image-Time-Series-Datasets?tab=readme-ov-file

[6] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997. https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext

[7] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734, 2014. https://arxiv.org/abs/1406.1078

[8] C. Pelletier, G. I. Webb, and F. Petitjean, *Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series*, Remote Sensing, vol. 11, no. 5, p. 523, 2019. https://www.mdpi.com/2072-4292/11/5/523

[9] H. I. Fawaz et al., *InceptionTime: Finding AlexNet for Time Series Classification*, Data Mining and Knowledge Discovery, vol. 34, no. 6, pp. 1435-1461, 2020. https://arxiv.org/abs/1909.04939

[10] B. Dang and Y. Li, *MSResNet: Multiscale Residual Network via Self-Supervised Learning for Water-Body Detection in Remote Sensing Imagery*, Remote Sensing, vol. 13, no. 16, p. 3122, 2021. https://www.mdpi.com/2072-4292/13/16/3122

[11] A. Vaswani et al., *Attention Is All You Need*, Advances in Neural Information Processing Systems, vol. 30, 2017. https://arxiv.org/abs/1706.03762

[12] J. Devlin et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186, 2019. https://aclanthology.org/N19-1423/

[13] K. He et al., *Deep Residual Learning for Image Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016. https://ieeexplore.ieee.org/document/7780459

[14] J. L. Ba et al., *Layer Normalization*, arXiv preprint arXiv:1607.06450, 2016. https://arxiv.org/abs/1607.06450

[15] E. J. Hu et al., *LoRA: Low-Rank Adaptation of Large Language Models*, International Conference on Learning Representations (ICLR), 2022. https://arxiv.org/abs/2106.09685