Self-supervised spatio-temporal representation learning of Satellite Image Time Series

Iris Dumeur, Silvia Valero, Jordi Inglada

Self-supervised spatio-temporal representation learning of Satellite Image Time Series

Iris Dumeur* *Student Member, IEEE*, Silvia Valero*, Jordi Inglada *
* CESBIO, Université de Toulouse, CNES/CNRS/INRAe/IRD/UPS 31000 Toulouse, France

***Abstract*—In this paper, a new self-supervised strategy for learning meaningful representations of complex optical Satellite Image Time Series (SITS) is presented. The methodology proposed named U-BARN, a Unet-BERT spAtio-temporal Representation eNcoder, exploits irregularly sampled SITS. The designed architecture allows learning rich and discriminative features from unlabeled data, enhancing the synergy between the spatio-spectral and the temporal dimensions. To train on unlabeled data, a time series reconstruction pretext task inspired by the BERT strategy is proposed. A Sentinel-2 large-scale unlabeled data-set is used to pre-train U-BARN. To demonstrate its feature learning capability, representations of SITS encoded by U-BARN are then fed into a shallow classifier to generate semantic segmentation maps. Experimental results are conducted on a labeled data-set (PASTIS). Two ways of exploiting U-BARN pre-training are considered: either U-BARN weights are frozen (named U-BARN^{FR}) or fine-tuned (U-BARN^{FT}). The obtained results demonstrate that representations of SITS given by U-BARN^{FR} are more efficient for land cover classification than those of a supervised-trained linear layer. Then, we observe in scenarios with scarce reference data-set that the fine-tuning brings a significative performance gain compared to fully-supervised approaches. We also investigate the influence of the percentage of element masked during pre-training on the quality of the SITS representation. Eventually, semantic segmentation performances show that the fully supervised U-BARN architecture reaches slightly better performances than the spatio-temporal baseline (U-TAE).**

***Index Terms*—Satellite Image Time series (SITS), Transformer, Self-Supervised Learning, Spatio-Temporal Network, Unet, Representation Learning**

I. Introduction

Over the last decade, the Satellite Image Time Series (SITS) acquired by the Sentinel-2 (S2) mission has produced a large amount of multi-spectral land surface imagery with a high 5-day revisit rate. The high spectral, spatial, and temporal resolutions of SITS capture physical measurements of temporal and spatial variations of the surface, making them crucial data for Earth monitoring [1],[2], [3]. Deep learning (DL) holds a great potential for automatically extracting features from spatio-temporal remote sensing data [4], [5]. Nonetheless, there are still significant challenges that DL architectures face in dealing with the particularities of SITS, which are non-stationary, multi-variate, and irregularly sampled. Data gaps induced by cloud contamination and data quality issues lead to a significant lack of information between optical valid acquisitions. In addition, undetected clouds can produce misleading results in land surface analysis. Besides the challenges associated with complex satellite data, DL methodologies in large-scale remote sensing applications face a major bottleneck. The limited availability and quality of the labeled data restrain the training of deep complex models. Over the past few years, self-supervised learning (SSL) has emerged as a potential solution to mitigate or even eliminate the need for costly collection of labeled data-sets [6]. This strategy enables the pre-training of deep models on large unlabeled data-sets for later fine-tuning a shallow network on a downstream task. Therefore, self-supervised pre-training methods can be a solution for applications collecting small labelled data-sets, where deep models cannot be trained from scratch.

Recent reviews [7],[6] have highlighted the great opportunities of self-supervised learning for remote sensing applications. Despite proposing different taxonomies, these studies agree that most of the proposed methods are based on discriminative models. In contrast, generative models such as GAN [8] and variational auto-encoders [9] that learn the latent distribution generating the input data have been less studied. This can be explained by the fact that latent variables capturing the distribution of observed variables cannot guarantee generalization capabilities for downstream tasks [10]. Among discriminative self-supervised learning studies, two main categories have been identified: contrastive approaches and methodologies using pretext tasks. Contrastive learning methods rely on data augmentation techniques that apply multiple transformations to the data without affecting their semantics. Although augmentation techniques have been defined for single satellite images as [11], [12], the augmentation of multi-spectral time series is not trivial. For this reason, existing contrastive methods exploiting Sentinel data mainly focus on optical and radar data, treating each modality as a distinct augmentation of the same object. For example, [13] processes pairs of single S1 and S2 images, while [14] handles pairs of S1,S2 SITS. However, it should be noted that this latter contrastive approach on SITS to pre-train deep architecture is not unsupervised, as classification labels are utilized to generate positive and negative samples required for the contrastive loss. Consequently, self-supervised training strategies based on pretext tasks are preferred on temporal data. This approach involves defining a task that can be solved using the input data alone, without the need for explicit labels. By generating a supervised learning strategy through pretext tasks, meaningful features can be extracted from the data. As an example, generative-based pretext tasks attempt to learn the structure of the data by posing a reconstruction task to recover

This work is supported by the DeepChange project under grant agreement ANR-DeepChange CE23

the features and information of the data itself. For instance, BERT [15] aims to recover masked words, and MAE [16] recovers masked pixels of images. Despite generative-based pretext tasks being one of the most promising strategies to exploit complex SITS, only two recent works are proposed in the literature [17] [18]. This can be explained by the strong challenges associated with : (i) the design of network architectures exploiting the complex SITS, and (ii) the pretext-task definition, ensuring that the learned representations are useful for downstream applications.

Considering all the above, this paper presents a novel self-supervised learning method for capturing meaningful representations of complex optical Satellite Image Time Series. The proposed methodology, named **U-BARN**, proposes a self-supervised learning strategy to learn a **U**net-**B**ERT sp**A**tio-temporal **R**epresentation e**N**coder. The first important contribution is the design of a new DL architecture that captures the spatio-temporal information contained in irregularly sampled multi-variate SITS. The spatial, spectral and temporal dimensions of the data are handled by the combination of Unet and Transformer architectures. Instead of using a traditional CNN [18], a Unet architecture is proposed to embed the spatio-spectral information by exploiting different spatial resolutions. By preserving the spatial input data dimensions, the Unet leads to highly efficient inference times. Compared with the most recent supervised end-to-end architecture [19], **U-BARN** proposes to apply temporal attention mechanisms at high spatial resolution, to capture more precise spatio-temporal information. The second significant contribution of this study is the self-supervised training of **U-BARN** which allows learning high-quality latent representations without requiring annotated data. Based on BERT [15], a generative pretext-task masking strategy is proposed. The general framework presented in this work is summarized in Fig. 1. On the left, we show the main blocks describing the backbone network of U-BARN, which is described in III-A. On the right, the use of pre-trained U-BARN in the semantic segmentation downstream task is illustrated.

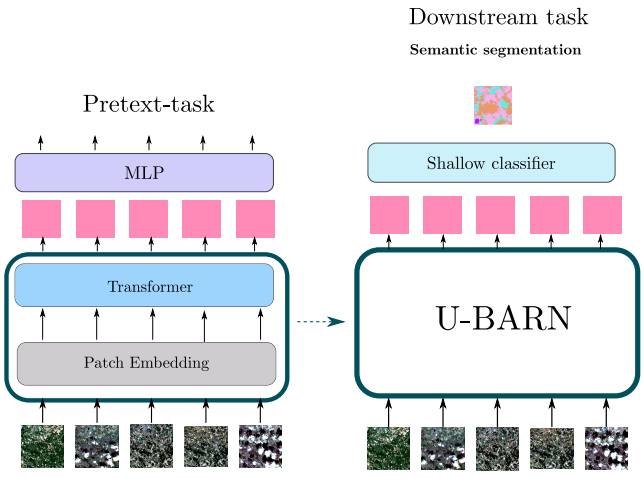

Figure 1. Left: Description of the proposed SSL strategy using BERT. Right: Description of how representations are used for the downstream semantic segmentation task.

To evaluate the performance of the proposed U-BARN architecture and the self-supervised training strategy, we conduct several experiments using the semantic segmentation downstream task defined by the labeled PASTIS data-set [19]. First, the pre-trained U-BARN segmentation performances are compared with two end-to-end trained architectures (U-TAE and U-BARN). Then, the usefulness of U-BARN is assessed by conducting several experiments on real-world scenarios suffering from scarce reference data. Lastly, to evaluate the impact of the pretext task, different experiments are carried out to study the influence of the complexity of the pre-training task on the quality of the spatio-temporal representations. The remainder of the paper is organized as follows: (i) a presentation of current state-of-the-art spatio-temporal architectures for SITS and existing SSL strategies are presented in Section II, (ii) a detailed description of our methodology is given in Section III, (iii) an explanation of the experimental setup is detailed in Section IV, (iv) the results obtained from the different experiments are presented in Section V, and finally, conclusions are drawn in Section VI.

For reproducibility, the large unlabeled S2 L2A data-set used to pre-train U-BARN as well as the code[1] will be available upon publication.

II. Related works

This section reviews: (i) the existing DL spatio-temporal architectures proposed to exploit SITS in a supervised way and (ii) SSL methods using pretext-tasks for temporal data.

A. Deep spatio-temporal architectures for SITS

Spectro-temporal patterns from multi-temporal data provide the most essential information to characterize land cover classes. For this reason, the earlier DL architectures exploiting recent SITS have not considered the spatial dimension of the data. For instance, TempCNN [20], which applies convolution on the temporal dimension, or Recurrent Neural Networks [21], [22], [23], [24], [25] which retain past timestamps information in memory, have been proposed. Although these architectures can outperform traditional approaches such as Random Forest [26], existing literature [27], [28], [29], has corroborated that better results could be obtained by also considering the spatial dimension. This is due to the fact that high-level spatio-temporal features allow the detection and discrimination of closely resembling spectral signatures. Convolutional Neural Networks (CNN) exploiting the spatial domain of SITS have been typically combined with temporal networks. For instance, the combination of CNN and RNN is proposed in [29], where the ReCNN architecture is introduced. The proposed network marries CNNs and RNNs as separate layers and the CNN output is injected as the input to an RNN. Other CNN and RNN combinations are proposed in [27] [28]. Both studies propose an architecture composed of two parallel branches aiming to independently extract spatial and temporal features. After the feature extraction step, the results of both branches are concatenated and injected in a

[1] https://gitlab.cesbio.omp.eu/dumeuri/ssl_ubarn

fully connected network to predict the final class. In M^3-fusion [27], the architecture proposes the fusion of Sentinel-2 (S2) pixel time series with Spot 6/7 VHSR patch images centered on the pixel of the time series. Features from temporal data are extracted by applying a RNN architecture whereas spatial features are learned by a CNN network applied on a high spatial-resolution 25×25 patch image. Although two parallel branches are also proposed in Duplo [28], this architecture exploits temporal S2 patches with a spatial dimension of 5×5 on both branches. The temporal branch uses a shallow CNN to reduce the spatial dimension to 1 before applying Gated Recurrent Units. The independent spatial branch process the temporal S2 patches by a more complex CNN architecture. This last study demonstrates that the combination of both network branches outperforms either CNN or RNN trained individually. However, the combined CNN-RNN architectures, [28], [27], [29] suffer from significant limitations when applied to SITS: (i) a narrow spatial neighborhood is considered, with a square patch width of only 50 meters (ii) inference is costly, since only the class of the center pixel within the patch is predicted. Alternative spatio-temporal architectures apply 3D CNN to learn the local temporal features along with the spatial ones [30] [31]. These latter architectures process inputs with wider spatial dimensions and [30] fully convolutional architecture is efficient for segmentation map prediction. However, only short temporal dynamics of the time series are learned by such architectures.

Additionally, the use of the aforementioned temporal architectures on SITS suffer from important weakness. First, RNN and TempCNN do not handle irregularly sampled time series, which implies that all SITS are first resampled to a common gap-free temporal grid. Secondly, long-term temporal dependencies are not fully captured, whereas correlation in temporal information between the beginning and the end of the annual SITS can be important.

To overcome the limitation of TempCNN and RNN architectures, the work in [32] propose to apply the Transformer network [33] in the spectro-temporal domain to classify S2 time series. This architecture (see Section III-A2) is applied on individual S2 pixel time series to extract spectro-temporal features for crop classification. Thanks to its attention layers and positional encoding, this architecture allows capturing relations between all the elements of a sequence and process irregular time series. The Transformer architecture also demonstrates cloud-robustness [32] compared to other architectures such as Duplo [28] and TempCNN [20]. The study in [32] shows that the Transformer is capable of identifying cloudy dates as outliers with low attention score. Recently, several transformer-based models are proposed for tackling SITS classification capturing temporal [17], [34], [35], [36], and spatio-temporal features [18], [19]. First, temporal approaches as [34][35] propose different solutions to reduce the high computational complexity of the classical Transformer network [33]. Both spectro-temporal models simplify the architecture by reducing the number of operations required to compute the attention score. The modified transformer, TAE described in [34] proposes to compute a unique master query to squeeze each individual pixel time series into a single embedding in the time dimension, which summarizes the global temporal information. A simplified version of TAE [34] is proposed by L-TAE [35], where the master query is set as a network parameter. This last architecture outperforms TempCNN [20], [32], as well as architectures with RNN, Conv-LSTM [37] and Conv-GRU [38]. As the altered attention mechanism focuses on global attention, [36] proposes a two branch temporal network GL-TAE, where the LTAE and the Lightweight convolution networks (LConv) respectively compute global and local attention. TAE, LTAE and LConv mechanisms squeeze the temporal dimension of the time series to 1, preventing the succession of multiple temporal encoder layers. To leverage the spatio-temporal dimensions of SITS, the SITS-Former [18] combines a three-dimensional CNN with a traditional Transformer. However, similarly to [28], [27], [29], a narrow spatial-context (i.e. patch size of 5×5 pixels) is considered and only the pixel at the center of the patch is classified. Alternatively, the U-TAE network [19] combining the L-TAE with a Unet network [39] has been recently proposed. The use of a Unet offers some advantages with respect to classical CNN architectures. By using contracting and expansive paths with skip connections between them, Unet features enable more accurate localization. Besides, larger receptive fields can be obtained by increasing the Unet depth, which allows extracting more context-rich spatial relationships. The U-TAE network [19] proposes to incorporate the L-TAE network within the Unet bottleneck. Although this choice considerably reduces the method's computational complexity, it implies that the temporal attention is only computed at the coarsest spatial resolution. Consequently, the ability to model temporal patterns can be reduced due to the encoder output resolution, which can lead to less accurate results.

Consequently, our proposed methodology, U-BARN, combines a Unet with a Transformer to capture rich and wide spatial and temporal correlations. The temporal attention mechanism is computed at a full spatial resolution. Therefore, our network produces embeddings which contain rich temporal information at the spatial resolution of the original data, which is expected to benefit downstream tasks like semantic segmentation.

B. Using self-supervised pretext tasks for temporal data

Self-supervised pre-training for sequence data has become hugely popular in Natural Language Processing (NLP). Most of existing techniques have used predictive or generative pretext tasks to capture temporal patterns from the data itself. Predictive strategies have proposed temporal shift prediction [10] or retrieving the order of a shuffled sequence [40]. In contrast, methods based on generative pretext tasks have learned to regenerate the input time series [41] based on some limited view of the data. Note that generative pretext tasks differ from generative models, which learn implicit distributions that allow to sample new data. The reconstruction of masked tokens (e.g. embedded words or sub-words) was shown to be an effective generative pretext task in NLP. More precisely, the BERT strategy proposed in [15] has become a de facto standard strategy to train a language representation model. In this strategy, a bidirectional Transformer backbone encoder is

trained to reconstruct input data by using information from tokens located both before and after the missing content. The excellent performance of BERT has led to the proposal of two similar generative pretext tasks in remote sensing [17] [18]. To the best of our knowledge, SITS-BERT [17] was the first self-supervised strategy exploiting SITS. This last study proposed to learn spectro-temporal features from Sentinel-2 by training a Transformer architecture. Specifically, a denoising pretext task goal is presented by simulating abnormal reflectance values caused by clouds, snow/ice and shadows. The corruption is obtained by adding positive or negative noise on a few dates. Following the same strategy, SITS-Former [18] was proposed by the same authors to learn more complex spatio-temporal features from multi-temporal data. Compared to [17], a more complex pretext task was proposed by SITS-Former by masking input patches with random values drawn from a normal distribution. The fine-tuned SITS-Former model showed impressive results for land cover classification tasks outperforming other models such as Random Forest, Duplo [28], SITS-BERT [17] and Conv-RNN [23]. As mentioned in the previous section, being not fully convolutional, SITS-Former can be highly inefficient to produce classification or segmentation maps. Besides its architectural limitations, the pretext task proposed by SITS-Former suffers from other limitations. First, SITS-Former uses the original masking rate proposed by BERT. Retrieving a masked word in NLP requires a holistic understanding of the sentence. However, in SITS, the continuity of spectral measurements usually allows the reconstruction of the missing input by simple interpolation. While some dates in SITS may be invalid due to the presence of clouds, shadows, or saturation, the masking rate may need to be adjusted to ensure that the pretext task is difficult enough. Secondly, distribution shift can significantly impact fine-tuning performance. In this context, distribution shift means that, at inference time, the data is not masked, and therefore, the distribution is different from the training data. To mitigate this effect, the original BERT employed an 80-10-10 strategy among the 15% of masking rate. Specifically, 80% of the masked words were replaced by the [MASK] token, 10% were left unchanged, and 10% were replaced by a random token value. However, as satellite data cannot be represented in a finite and discrete embedding space like natural language, the choice of mask values should differ from NLP. While SITS-Former proposed masking only with random values drawn from a normal distribution, this approach does not adequately address the distribution shift issue. Thirdly, while [32] has demonstrated that Transformer attention networks can handle invalid acquisitions, SITS-Former is exclusively trained on cloud-free SITS. Therefore, the self-supervised strategy employed by SITS-Former may not perform well on downstream tasks that involve non-filtered or imperfectly filtered cloud data.

III. Proposed methodology

This section presents the network architecture of the U-BARN encoder and the proposed pre-training strategy.

A. U-BARN network architecture

The U-BARN backbone network is mainly divided in two main blocks: (i) the patch embedding layers providing a spatio-spectral representation of each independent image patch of the time series and (ii) the transformer block capturing the temporal relations between the patch embeddings of the time series. U-BARN generates spatio-temporal SITS representations, at the same spatial and temporal resolutions than the input SITS. Specifically, given a batch of input patch time series (B, T, C, H, W) with B the batch, T the temporal, C the spectral, and H, W the spatial dimensions, U-BARN generates a batch of patch time series representations (B, T, d_{model}, H, W) with d_{model} the number of features.

1) Patch embedding: As shown in Fig. 2, this block embeds each patch of the time series with its corresponding positional encoding. Considering a time series of T dates, the spectro-spatial encoder (SSE) independently encodes each patch into feature map. As a result, patches of dimension (C, H, W) are projected in to feature vectors of size (d_{model}, H, W). The proposed SSE is based on a Unet architecture with four down-sampling and up-sampling levels as shown in Fig. 2. This Unet implementation enables to capture high-level spatial features with a wide field of view. For each down-sampling and up-sampling level, the spatial dimension of the feature map is respectively divided and multiplied by 2. the Unet architecture is similar to the U-TAE [19] although the temporal attention mechanism is removed from Unet bottleneck. As no temporal dimension is exploited in the SSE, input time series (B, T, C, H, W) are reshaped to $(B \times T, C, H, W)$, before being processed by the Unet. We expect that during training the SSE learns to generate, for each pixel, features which contain spectral information as well as rich and wide spatial context.

To incorporate temporal information (relative and absolute ordering) of the original times series on the learned SSE feature maps, the classical positional encoding [33] is added to each encoded patch of size d_{model}. As denoted by Eq. 1, the strategy uses sine functions of varying frequencies for even embedding indexes ('i') and cosine functions for odd embedding indexes. The term i refers to each of the d_{model} features. As proposed by [17], the acquisition day of year (DOY) of each image is used to indicate the position of the patches in the time series. As recommended in [34] a scaling constant of a 1000 is considered.

$$PE(DOY, 2i) = \sin(\frac{DOY}{1000^{2i/d_{model}}}) \tag{1a}$$

$$PE(DOY, 2i+1) = \cos(\frac{DOY}{1000^{2i/d_{model}}}) \tag{1b}$$

2) Transformer block: This network architecture aims to exploit temporal relations of the series of feature maps resulting from the patch embedding layers (see Fig. 1). Under this goal, each time series of features describing a single pixel is individually processed by the Transformer architecture. Considering that, the dimension of the batch of pixel-level time series fed in the network is equal to $(B \times H \times W, T, d_{model})$. The

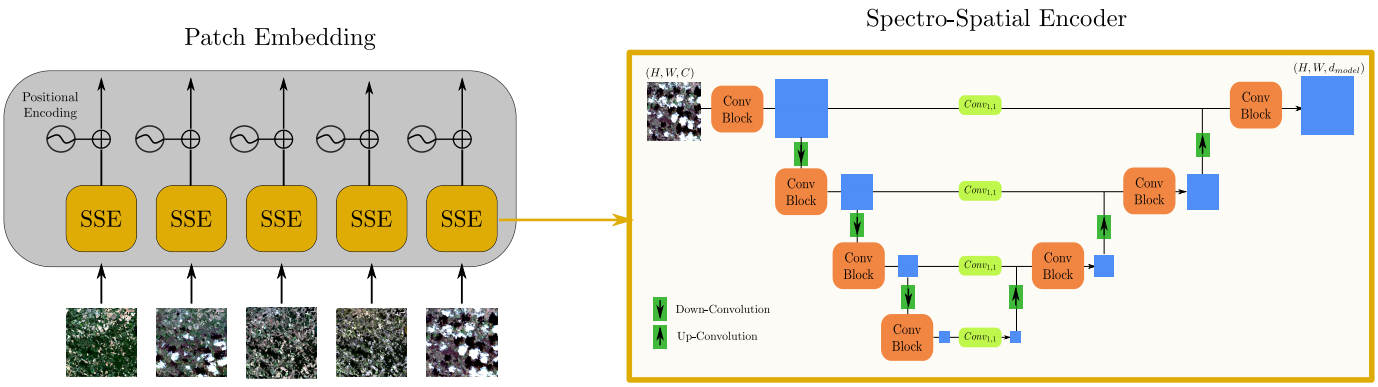

Figure 2. Left: Overview of the patch embedding. A spatial-spectral encoder (SSE) embeds each patch into a (H, W, d_{model}) feature map. A positional encoding is added on the resulting feature maps. Right: Detailed description of the SSE architecture.

backbone network is composed of multi-head self-attention and feed forward layers, as detailed in Fig. 3.

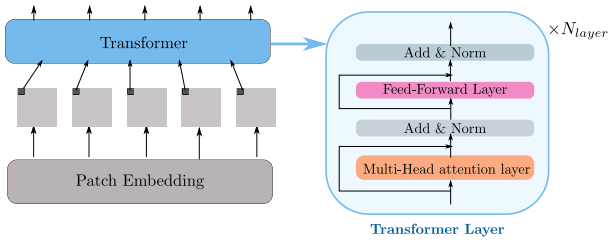

Figure 3. Overall architecture of the spectro-temporal encoder. The Transformer processes pixel-level time series.

The multi-head attention module decomposes the attention in multiple heads running in parallel as illustrated in Fig. 4. Each head is composed by an attention mechanism which computes similarity scores for all pairs of positions in a pixel-level time series. These scores are computed by applying a scaled dot product operation on the q and k representations of an input time series X as described by Eq. 2. These representations denoted by "query", $q = W_Q X$ and "key" $k = W_K X \in \mathbb{R}^{T * d_{model}}$ are obtained by the learned projection matrices W_Q and W_K. As denoted by Eq. 2 and illustrated in Fig. 4 the dot product result is passed through a softmax operation. The resulting scores then weight another representation of the input time series, called "value" $v = W_V X$. These weights give indication on which acquisitions are important for the training task.

$$Attention(q, k, v) = \text{softmax}(\frac{qk^T}{\sqrt{d_{model}}})v \quad (2)$$

As demonstrated in [33], the computation of scaled-attention products on different feature subspaces allows each attention head to focus on different features leading to better performances and training stability. Accordingly, instead of computing one scaled-dot product on a unique set of query q, key k and value v, the input X is split along the feature dimension into H subvectors. Scaled-dot product is computed in parallel on H triplets, called "heads" as depicted in Fig. 4. The resulting H time series are then concatenated and fed into

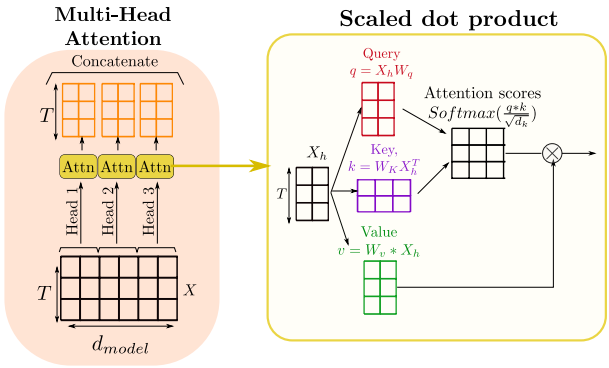

Figure 4. Left: Description of the multi-head self-attention mechanism on a sequence of dimension (T,d_{model}). Right: Scaled-dot product on time series.

feed-forward layers that operate only on the feature (spectral) dimensions.

Feed forward layers are composed of two linear layers interspersed with a ReLu activation layer. Inside this feed-forward block the first fully-connected (FC) layer projects the features into d_{hidden}-dimensional space, while the second FC layer projects the feature maps into d_{model}-dimensional space.

Theoretically, increasing the number of layers and the number of heads improves the quality of the learned representation. Therefore, the U-BARN transformer block is composed of 3-layers (as [17]) with each 4 heads. The dimension of input and output features of the network are respectively set to $d_{model} = 64$ and $d_{hidden} = 128$. The architectural hyper-parameters are detailed in Annex (see Appendix A).

B. Self-supervised strategy

Fig. 5 shows the overall framework of the proposed self-supervised pre-training strategy inspired by the BERT [15]. As observed, the proposed pretext task aims to reconstruct some input patches that have been masked from the original time series. Specifically, the masking step is randomly applied on SSE output representations and a decoder MLP network is used for the inpainting task.

1) Masking strategy: Two parameters are required for the masking process: the percentage of data to be masked and the masking values used to substitute original embedding

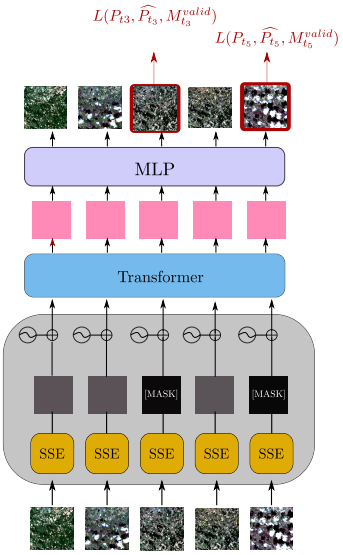

Figure 5. Description of the proposed self-supervised strategy. A percentage M_{rate} of the feature maps encoded by the SEE are masked. U-BARN is trained to reconstruct the previously corrupted patch. The reconstruction loss is computed on the valid pixels, given by the binary mask M^{valid} associated to the masked patch.

representations. Given an input time series, M_{rate} corresponds to the percentage of masked timestamps to be reconstructed. To increase the diversity of training samples, the masked timestamps are drawn randomly for each time series and vary at each training epoch. The masking values used to corrupt the data can introduce outliers or unrealistic values that do not exist in downstream tasks. This phenomenon, known as *distribution shift*, is prevented in U-BARN masking strategy. The proposed strategy consists in randomly permuting the spectro-spatial embedding values among selected encoded patches within a batch. Specifically, an embedded pixel can be replaced by an embedded value from another date, another pixel location within the batch or another feature.

2) Decoder: The decoder, which is only used for pre-training and discarded afterward, enables to train the U-BARN in a self-supervised way. As shown in Fig. 1, the decoder reconstructs the input data using the latent representation.

In order to avoid leakage of meaningful features in the pre-training decoder, a very simple and shallow decoder composed of a single linear layer is proposed. The decoder operates exclusively on the feature (spectral) dimension to transform the (B, T, d_{model}, H, W) latent representations into the S2 reconstructed patch time series (B, T, C, H, W).

3) Reconstruction loss: The quality of the reconstructed image patches is evaluated during the training by the classical Mean Square Error (MSE). This reconstruction loss is computed exclusively on the corrupted patches. Moreover, as input patches can have invalid measures due to acquisition conditions (e.g. cloudy and out of swath pixels), the information coming from the valid acquisition mask M^{valid} is incorporated in the loss function. Therefore, invalid input pixels belonging to the corrupted input patches are not considered in the reconstruction loss. Eventually, given an input patch time series $[P_{t_1}, ..., P_{t_{L\,max}}]$, a set T_S of masked dates, $N_{t_k}^{valid}$ the number of valid pixels in the patch P_{t_k}, the resulting loss can be expressed as Eq. 3. $\widetilde{P_{t_k}}$ is a patch that is corrupted at the SSE output by the previously described masking strategy.

$$L = \frac{1}{||T_S||} \sum_{t_k \in T_S} \frac{M_{t_k}^{valid}}{N_{t_k}^{valid}} \odot ||UBARN(\widetilde{P_{t_k}}) - P_{t_k}||_2^2 \quad (3)$$

IV. Experimental setup

First, the two Sentinel 2 L2A data-sets used in our different experiments are presented: the unlabeled large scale data-set used for pre-training U-BARN and the downstream labeled PASTIS data-set.

Secondly, the implementation details of our downstream tasks are described.

A. Data-Sets

Sentinel-2 images processed to surface reflectances (L2A) by Theia are used in this study. For these data-sets, only the four 10 m and the six 20 m resolution bands of S2 are used. The 20 m resolution bands are resampled onto the 10 m resolution grid by bi-cubic interpolation. A robust data normalization is applied on S2 L2A reflectances. First, the scaling technique of Eq. 4a using the 0.05 and 0.95 quantiles is applied to remove data outliers by clipping the data. Secondly, the data is centered by subtracting the median value of x_{clip} to each spectral band and dividing the result by the dynamic data range (see Eq. 4b)[2]. Band statistics used to normalize the two Sentinel-2 data sets are computed on the large unlabeled pre-training training data-set.

$$x_{clip} = clip_{q_{0.95}, q_{0.05}}(x) \quad (4a)$$

$$x_{norm} = \frac{x_{clip} - median}{q_{0.95} - q_{0.05}} \quad (4b)$$

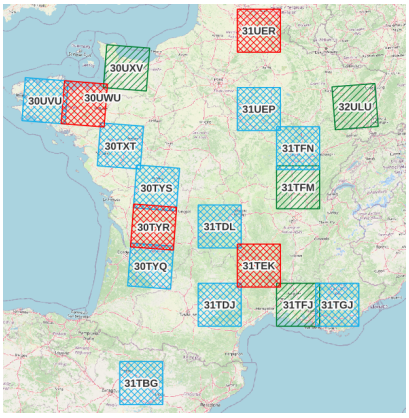

Figure 6. Description of the S2 data-sets used for pretext and downstream tasks. The unlabeled data-set for pre-training is composed of two disjoint data-sets: training (tiles in blue) and validation (tiles in red). S2 tiles in the labeled PASTIS data-set are shown in green.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

1) Large-scale unlabeled pre-training data-set: The data set is composed of 14 tiles acquired by Sentinel-2 over France and Catalonia (Spain). The corresponding validity masks (non-corrupted pixels) are built by considering edge, saturation and cloud information. As previously explained, the information contained in validity masks is incorporated in the reconstruction loss of the pretext task. Geographical variability between training and downstream task is enforced by using disjoint tile sets between the PASIS data-set and the unlabeled data-set, as shown in Fig. 6. We have more diverse pre-training data-set, compared to that of the SITS-Former [18] data-set, which is only composed of SITS from 3 S2 tiles from 2018 or 2019. The U-BARN pre-training is performed by considering 10 different S2 tiles acquired from 2018 to 2020. In each of these tiles, 10 smaller regions of interest (ROIs) of size 1024×1024 are randomly selected. During training, to build patch time series of spatial dimension (64 × 64), a random crop[3] is operated on the 1024×1024 ROIs. As a result, diverse samples are processed by U-BARN during pre-training. The disjoint validation data set is composed by the 4 remaining S2 tiles acquired from 2016 to 2019. For each year, 10 patch time series, of spatial dimension (64× 64) are extracted from each of the 4 tiles and used to tune the hyper-parameters. The validation data-set is used to select the best model weights, which are then used for the PASTIS downstream task. A more exhaustive description of the unlabeled data-set is given in Table I

Table I
PRE-TRAIN DATA-SET DESCRIPTION

Data-Set	S2 tiles	Year
train	T30TXT, T30TYQ, T30TYS, T30UVU, T31TBG, T31TDJ, T31TDL, T31TFN, T31TGJ, T31UEP	2018-2020
val	T30TYR, T30UWU, T31TEK, T31UER	2016-2019

2) PASTIS: This labeled Sentinel-2 data-set proposed for semantic segmentation in [19] covers agricultural areas over France as shown in Fig. 6. Based on the French Land Parcel Information System, the agricultural parcels are grouped into 18 different crop classes. Although PASTIS contains SITS acquired from September 2018 to November 2019, only data from January 2019 to November 2019 is considered in our experiments. This requirement is imposed by our pre-training data-set which is composed of annual time series.

The complete data-set contains 2433 patch time series, and it is divided into 5 stratified folds to enable k-fold training. Therefore, to train the model on the PASTIS data-set, 5 trainings will be performed. In each of these experiments, 3 folds are attributed to train data, one for validation purpose and the last one for testing (see Table II).

The spatial dimension of PASTIS patch times series is equal to (128×128). Therefore, a random crop transformation is operated during training to obtain the spatial dimension of (64× 64) used in the pre-training stage. For validation and testing, there should be no randomness in the spatial crop, therefore center crop transform[4] is applied on the patch time series.

Table II
OFFICIAL 5-FOLD CROSS VALIDATION SCHEME GIVEN BY [19]

Fold	Train	Val	Test
I	1-2-3	4	5
II	2-3-4	5	1
III	3-4-5	1	2
IV	4-5-1	2	3
V	5-1-2	3	4

B. Details of the downstream task implementation

In the downstream semantic segmentation task, the reconstruction decoder described in Section III-B2 is replaced by a shallow classifier (SC) as shown in Fig. 1. The objective of the classifier is to generate segmentation maps from the latent representations encoded by U-BARN. The selection of the architecture of the SC is driven by the two following criteria. First, the U-BARN encoder produces latent representations preserving the temporal size of the input time series. Therefore, the classifier should be able to process inputs with different temporal dimensions. Secondly, since this is a segmentation task, the output of the shallow classifier should have no temporal dimension.

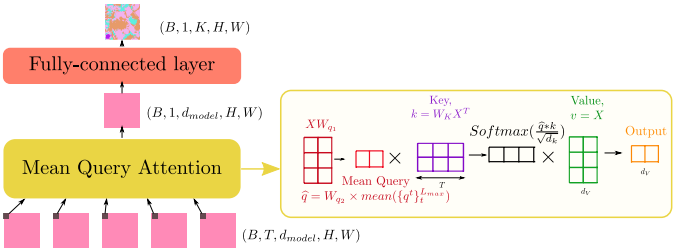

Figure 7. Architecture of the shallow classifier and detailed description of the "mean-query" attention mechanism described in [34]

To meet both requirements, we have designed a shallow classifier (SC), shown in Fig. 7. To process input with different temporal dimension, the proposed SC utilizes the mean-query attention mechanism proposed in the TAE [34]. In this altered attention mechanism, a master query, which is the temporal average of the queries, is computed. Additionally, in the computation of the "value" representation, the time series X is not projected by a matrix W_v, thus $v = X$ in Eq. 2 . As shown in Fig. 7, the output of this mean-query attention has a collapsed temporal dimension. The mean-query attention mechanism followed by a Fully-Connected (FC) layer, to project the $(B, 1, d_{model}, H, W)$ feature map into the $(B, 1, K, H, W)$ segmentation map, with K the number of classes. As suggested in [19], the cross-entropy loss is exclusively computed on known crop classes.

[3]https://pytorch.org/vision/main/generated/torchvision.transforms.RandomCrop.html

[4]https://pytorch.org/vision/main/generated/torchvision.transforms.CenterCrop.html

C. Training scenarios evaluated on the downstream tasks

According to [6], to evaluate self-supervised tasks, *linear-probing* and *fine-tuning* are often operated. Traditionally, the linear probing strategy evaluates the representations by a linear classifier which is trained on top of a learned and frozen encoder. Unfortunately, a linear classifier can not be applied on U-BARN latent spaces since the temporal length of the resulting U-BARN time series representations varies for each patch time series. The linear classifier is thus replaced by the SC, presented in Section IV-B, and it is trained to generate maps from representations obtained by a frozen pre-trained U-BARN encoder. This method, referred as U-BARN^{FR}, enables to drastically reduce the number of training weights in the downstream task, as solely the SC is trained. For the fine-tuning approach, the weights of the U-BARN encoder are not frozen during the training of the downstream task. However, the weights of the pre-trained U-BARN are used as the starting values for training of the complete architecture. The fine-tuning strategy is denoted by U-BARN^{FT}. To assess the quality of pre-trained U-BARN models, the previous self-supervised scenarios are compared with three training configurations supervised by the PASTIS data set. The first one is denoted by U-BARN^{e2e} and it corresponds to a trained end-to-end U-BARN encoder followed by the SC. The U-BARN^{e2e} encoder can be considered as the U-BARN^{FR} **higher bound** since frozen model performances are not expected to surpass its end-to-end counterpart. In contrast, it is expected that U-BARN^{FT} outperforms the U-BARN^{e2e} model which is trained from scratch. The quality of representations obtained by the pre-trained U-BARN models are also evaluated by a **lower-bound**. The idea is to compare the features learned by U-BARN with representations encoded by a single fully connected layer. For this situation U-BARN is replaced by a FC layer, which operates exclusively on the feature (spectral) dimension. The FC layer increases the spectral dimension (10 spectral bands) to d_{model}. This lower bound is obtained by the end-to-end supervised training scenario denoted by FC-SC.

Finally, the supervised spatio-temporal baseline U-TAE [19] is also considered in our experiments.

V. Experiments and Analysis

In this section, the proposed U-BARN network architecture and the self-supervised training strategy are evaluated by the PASTIS segmentation downstream task. First, a qualitative evaluation of the pretext task training is proposed. Then, the quality of the representations learned by pre-trained U-BARN models are evaluated by comparing the PASTIS classification performances obtained by the aforementioned different training scenarios (see Section IV-C). The interest of using a pre-trained U-BARN self-supervised encoder is corroborated by studying the robustness of the proposed methodology under reference data scarcity conditions. Finally, the influence of the masking rate on the generalization capabilities of U-BARN representations is studied.

Each training (either pre-training or downstream task) involves training the networks for a minimum of 100 epochs. The learning rate is set to 0.001, and a learning rate on plateau reduction scheduler is used with a patience of 10 epochs. The networks are trained on a single GPU, which could be a Tesla V100, A100, or A30, with a batch size of 2.

A. Qualitative assessment of the pre-training

This section presents an analysis of U-BARN's performance on the pre-training task. To evaluate the effectiveness of U-BARN on this task, we examine some reconstructed patches from the unlabeled validation set, as shown in Fig. 8. The results demonstrate that U-BARN is able to reconstruct the temporal evolution of masked continuous blocks of dates (e.g., DOY 72 to 102 and 142 to 175 in Figure 8). Therefore, we consider that U-BARN can successfully learn the temporal dynamic of the SITS during pre-training. Furthermore, Fig. 8 also shows that U-BARN reconstructs ground surface reflectances of cloudy patches (see DOY 102, 115 and 142). This result can be explained by the fact that the reconstruction of cloudy patches is not forced in the loss function (see Eq. 3). Following [32], we assume that the model learns that cloudy pixel values can be interpreted as outliers in the temporal profile. Under this situation, the network learns how to ignore their values for the patch reconstruction. Overall, our observations of U-BARN's performance on the pretext-task provide evidence that pre-training is successful, as U-BARN is able to effectively solve the pretext-task.

B. Classification performances on PASTIS data-set

The classification performances obtained by the above described training scenarios are compared here. The U-BARN model is pre-trained on the unlabeled dataset with the proposed generative pre-text task strategy. The pre-training stage considers a masking rate equal to 60% which is justified by the results described in Section V-D. Four different classification metrics are used to evaluate the quality of the obtained results : Cohen Kappa, overall accuracy (OA), F1 score and mean Intersection over union (mIoU). The two latter metrics are averaged per classes and not per pixel as the overall accuracy. As we proceed to 5-fold training with PASTIS, mean and standard deviation of the classification metrics are given each time. The overall results comparing the different training scenarios are reported in Table III. The F1-score per class is also given in Table IV to bring detailed information on the classification of each class in the unbalanced PASTIS dataset. Eventually, the confusion matrix, from U-BARN^{FR} and FC-SC, are shown in Fig. 9 and an example of the segmentation maps produced by the different networks is displayed in Fig. 10.

1) Frozen encoder U-BARN^{FR}: As observed in Table III and in Table IV, the performance of U-BARN^{FR} is intermediate between the FC-SC and the U-BARN^{e2e}.

Compared to the FC layer, the pre-trained and frozen U-BARN^{FR} obtain a gain in Kappa of 0.058, 0.042 in OA, 0.045 in F1-score and 0.048 mIou. The F1-score per class also highlights that the classification gain differs for each class, with a significant improvement (at least 0.3 in F1-score) for spring barley, potatoes, and orchards. We also observe a gain of at least 0.1 in F1-score, for winter durum wheat,

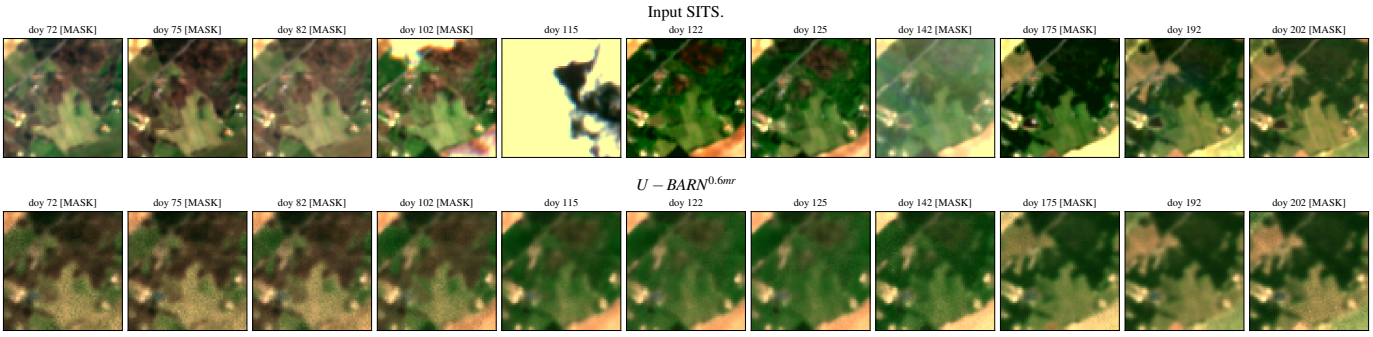

Figure 8. Example of a patch (from the validation data-set) reconstruction achieved by U-BARN during pre-training. Only a part of the SITS is displayed. DOY of each patch are indicated. [MASK] indicates that the embedded patch was corrupted (see Section III-B1). The top row is the input SITS, and the bottom row corresponds to reconstructions produced by U-BARN. During this pretraining the M_{rate} equals 60%.

Table III
CLASSIFICATION METRICS AVERAGE AND STANDARD DEVIATION OVER PASTIS K-FOLDS FOR DIFFERENT SITS ENCODER

	Kappa	OA	F1	mIoU
FC-SC	0.631 ± 0.015	0.770 ± 0.013	0.670 ± 0.011	0.284 ± 0.007
U-BARN^{FR}	0.689 ± 0.006	0.812 ± 0.009	0.715 ± 0.006	0.332 ± 0.002
U-BARN^{FT}	**0.831** ± 0.008	**0.901** ± 0.008	**0.841** ± 0.009	0.536 ± 0.008
U-BARN^{e2e}	**0.831** ± 0.008	**0.902** ± 0.007	**0.848** ± 0.010	0.539 ± 0.011
U-TAE	0.815 ± 0.010	0.893 ± 0.008	0.831 ± 0.008	**0.548** ± 0.007

Table IV
F1 SCORE PER CLASS ON PASTIS DATASET FOR DIFFERENT SITS ENCODER

	FC-SC	U-BARN^{FR}	U-BARN^{FT}	U-BARN^{e2e}	U-TAE
Meadow	0.780 ± 0.020	0.788 ± 0.024	**0.877** ± 0.021	**0.878** ± 0.021	0.863 ± 0.019
Soft winter wheat	0.750 ± 0.015	0.790 ± 0.018	**0.887** ± 0.019	**0.893** ± 0.009	0.878 ± 0.031
Corn	0.847 ± 0.012	0.867 ± 0.014	**0.934** ± 0.010	**0.939** ± 0.008	**0.932** ± 0.012
Winter barley	0.539 ± 0.027	0.718 ± 0.030	**0.895** ± 0.021	**0.901** ± 0.020	**0.893** ± 0.019
Winter rapeseed	0.856 ± 0.039	0.868 ± 0.019	0.948 ± 0.010	**0.954** ± 0.006	0.946 ± 0.015
Spring barley	0.209 ± 0.048	0.602 ± 0.066	**0.808** ± 0.056	**0.805** ± 0.048	**0.783** ± 0.035
Sunflower	0.599 ± 0.070	0.653 ± 0.037	**0.862** ± 0.034	**0.862** ± 0.029	0.829 ± 0.036
Grapevine	0.625 ± 0.044	0.680 ± 0.044	**0.862** ± 0.021	**0.858** ± 0.018	**0.853** ± 0.025
Beet	0.858 ± 0.032	0.873 ± 0.012	**0.953** ± 0.019	**0.948** ± 0.013	0.924 ± 0.027
Winter triticale	0.058 ± 0.042	0.191 ± 0.036	0.685 ± 0.033	**0.708** ± 0.015	**0.697** ± 0.057
Winter durum wheat	0.526 ± 0.043	0.644 ± 0.021	**0.785** ± 0.026	**0.782** ± 0.042	0.704 ± 0.083
Fruits, vegetables, flowers	0.201 ± 0.063	0.324 ± 0.023	**0.678** ± 0.030	**0.706** ± 0.039	0.636 ± 0.057
Potatoes	0.209 ± 0.048	0.532 ± 0.097	**0.760** ± 0.066	**0.742** ± 0.081	0.687 ± 0.112
Leguminous fodder	0.279 ± 0.073	0.250 ± 0.048	**0.628** ± 0.038	**0.638** ± 0.033	0.585 ± 0.029
Soybeans	0.645 ± 0.067	0.747 ± 0.037	**0.913** ± 0.021	**0.916** ± 0.022	**0.903** ± 0.035
Orchard	0.179 ± 0.023	0.520 ± 0.034	**0.698** ± 0.062	**0.703** ± 0.049	**0.681** ± 0.064
Mixed cereal	0.066 ± 0.038	0.081 ± 0.030	0.552 ± 0.034	**0.606** ± 0.034	0.564 ± 0.051
Sorghum	0.171 ± 0.071	0.150 ± 0.075	**0.599** ± 0.086	**0.620** ± 0.040	**0.589** ± 0.051

soybeans, winter barley, and fruit vegetables & flowers. The classification matrices shown in Fig. 9 show that U-BARN^{FR} has fewer confusions than the FC-SC. Specifically, U-BARN^{FR} performs better at distinguishing sunflower from potatoes and fruit, vegetable and flowers. Compared to the FC layer encoding, U-BARN^{FR} also mitigates confusion between spring and winter barley. Therefore, we conclude that the representations provided by U-BARN^{FR}, compared to SITS encoded by a FC layer, contain meaningful and discriminative information for the shallow classifiers. Since U-BARN^{FR} outperforms FC-SC on all classification metrics, our self-supervised pre-training strategy is shown to be effective. However, the performance gap between U-BARN^{FR} and U-BARN^{e2e} suggests that there is still room for improvement. A visual inspection of the segmentation maps generated by U-BARN^{FR} (shown in Fig. 10) reveals an issue with spatial consistency. The appearance of classification noise can be attributed to the fact that the masking self-supervised strategy is mostly applied on the temporal domain. Therefore, the proposed pretext task does not allow to completely learn the spatial correlations between pixels. As U-BARN^{e2e} segmentation maps does not exhibit this same issue, we consider that this weakness is due to the pretext task and not the architecture itself.

2) Fine-tuning U-BARN^{FT}: The global classification metrics presented in Table III and the F1-score per class in Table IV show that there is little difference between the performances of U-BARN^{e2e} and U-BARN^{FT}. It appears that fine-tuning does not lead to any improvement in classification performance. We conjecture that the number and diversity of training labels available in the PASTIS data-set are sufficient to train the U-BARN^{e2e} model. This assumption is later investigated in Section V-C, where the classification performances of

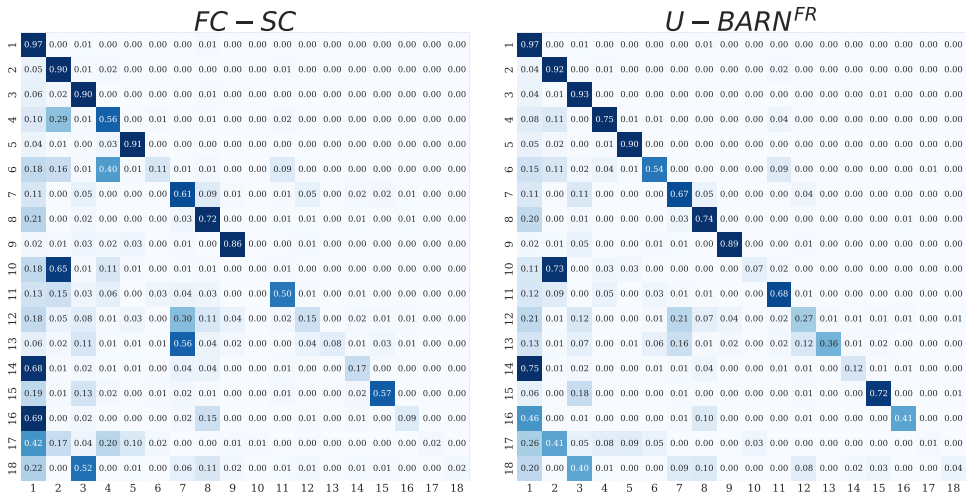

Figure 9. Confusion matrices on the PASTIS segmentation task. On each confusion matrix, rows correspond to true label and columns to predictions. The matrices are normalized per row. The correspondence between PASTIS classes and the confusion matrix index is the following: {1: Meadow, 2: Soft winter wheat, 3: Corn, 4: Winter barley, 5: Winter rapeseed, 6: Spring barley, 7: Sunflower, 8: Grapevine, 9: Beet, 10: Winter triticale, 11: Winter durum wheat, 12: Fruits, vegetables, flowers, 13: Potatoes, 14: Leguminous fodder, 15: Soybeans, 16: Orchard, 17: Mixed cereal, 18: Sorghum}

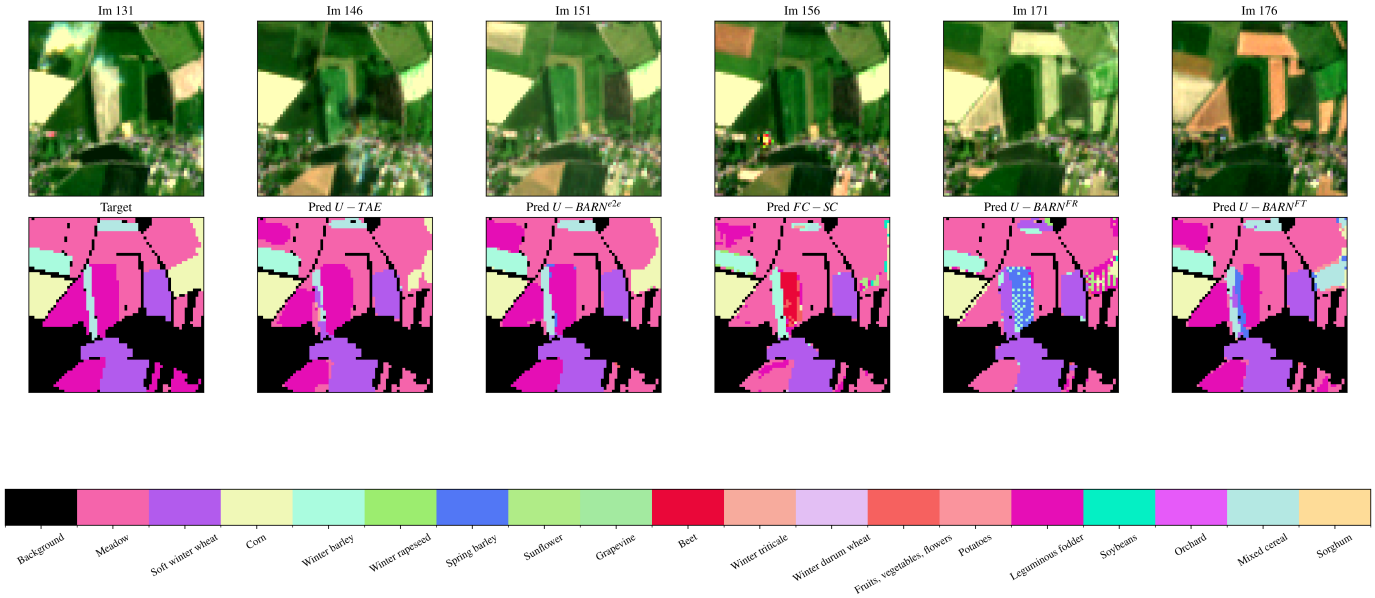

Figure 10. Top row, some of the S2 RGB images which belong to input time series. Bottom row, different segmentation maps generated by the different networks. From left to right: target segmentation map, U-BARN{e2e}, U-BARN^{FR}, U-BARN^{FT}, U-TAE and the FC-SC predictions.

both U-BARN models are compared in scenarios with scarce reference data.

3) U-BARN architecture: The U-BARN backbone network can be evaluated by comparing the metrics obtained by supervised U-TAE and U-BARN^{e2e} models. The results in Table III and Table IV reveal close performances for both models. Whereas the highest mIoU is obtained by U-TAE, U-BARN^{e2e} has a significantly higher F1 score, OA and Kappa. Looking more specifically at the F1 score per class, we notice that the performances slightly vary depending on the type of crop, as shown in Table IV. Among 10 of the 18 classes, U-BARN^{e2e} F1 score is significantly higher than that of the U-TAE. Eventually, as shown by the segmentation maps Fig. 10, U-TAE retrieve slightly worse edges than U-BARN^{e2e}. Contrary to our expectations, we did not find that on a crop classification task U-BARN^{e2e} totally surpass U-TAE. A reasonable explanation is that attention at full spatial resolution is not an important asset in the PASTIS crop classification task. In the PASTIS data-set, small crops labels are discarded and considered as background, resulting in no assessment of segmentation of small items. Additionally, it must be noted that the metrics found are slightly lower than those found in the original UTAE study [19]. This can be explained by the fact that the SITS used are temporally smaller, as we process annual SITS as detailed in Section IV-A2. As a conclusion, the overall results show that training the U-BARN architecture

by using an end-to-end supervised task has slightly better performances than the U-TAE [19] on PASTIS data-set.

C. Impact of the amount of training data on fine-tuned U-BARN models

In spite of satellite data being now available in abundance, ground truth reference labels remain scarce and costly to obtain. As demonstrated in [18], the performance gap between pre-trained SITS-Former and end-to-end trained models increases as the number of training labels decreases. Therefore, a similar experiment conducted on the PASTIS data-set is presented here. The goal is to compare the performances of U-BARN^{FT}, U-BARN^{e2e} and U-TAE models by reducing the size of training data-set. In this experiment, U-BARN^{FT} is pre-trained with a masking rate of 60%. As previously mentioned, the PASTIS data-set is divided into five folds. To simulate label scarcity, for each of the five experiments, we have randomly selected N_{SITS} patch time series from the three folds assigned to the training set. However, the PASTIS data-set exhibits a strong class imbalance. To ensure that all classes are present in the generated reduced training data-sets, the random selection of the patch time series follows the specific protocol detailed in Appendix B. Due to the small size of the resulting data-set, we have generated five smaller training data-sets, each composed of N_{SITS} SITS, for each training experiment. Finally, in this experiment, due to K-Fold training, we have conducted 25 trials to assess the performance of a pre-trained model with a training data-set composed of N_{SITS}. The different trials are used to compute the means and standard deviations of the classification metrics for the different models. Fig. 11 plots the metrics as a function of the number of training labels. With a training data-set composed of 30 patch time series, U-BARN^{FT} has a significantly higher Kappa and OA than U-BARN^{e2e}. The fine-tuning is therefore effective to boost performance when training with a reduced number of labels. Besides, on all the 4 classification metrics with N_{SITS} equals to 30 and 50, U-BARN^{FT} and U-BARN^{e2e} outperforms the U-TAE. We assume that because the U-TAE computes temporal attention at a low spatial resolution, the attention mechanism process fewer pixel time series than the U-BARN, and therefore is less competitive. On the Kappa, OA and F1 score curves, we see a similar trend: the gap between the U-BARN^{FT} , U-TAE and U-BARN^{e2e} performances reduces when N_{SITS} increases. These experiments corroborate previous results from SITS-Former [18]; as the number of samples increases, the performance gain, obtained thanks to pre-training, decreases. This experiment highlights the effectiveness of our approach in real-world scenarios with limited training labels.

D. Influence of the masking rate

Theoretically, the quality of the learned representations tends to improve when the pretext task becomes harder to solve (see Section II-B). Therefore, the experiment carried out here aims to investigate if a higher masking rate creates a harder and more meaningful pre-training task that can excavate deeper feature information. However, if this rate is set too high, the corrupted time-series become meaningless, making the task unsolvable. In this regard, we compared the performance of U-BARN^{FR} pre-trained with different M_{rate} values using the previously described classification metrics. The obtained results are shown in Fig. 12 and exhibit two local maximum for M_{rate} equals to 30 and 60%. This observation could be explained by the double effect of varying the masking rate in the pre-training. As the masking rate increases, the number of "valid" dates used to reconstruct the corrupted patches diminishes, and the reconstruction loss during pre-training is applied to more patches during each optimization step. Eventually, we consider that best performances are reached with M_{rate} 60%. This also suggests that the 15% masking rate proposed in NLP for BERT [15] may not be optimal for pre-training our spatio-temporal architecture with SITS. Additionally, results show that a masking ratio greater than 80% causes a significant drop in 3 out of 4 classification metrics (Kappa, OA and mIoU), indicating that the pretext-task might have become too difficult for training purposes.

VI. Conclusion

This paper proposes a novel self-supervised methodology for learning spatio-temporal representations from satellite image time series. The U-BARN architecture combines the strengths of Unet and Transformer to extract informative and discriminative features from unlabeled data-sets. Compared to U-TAE, which is the current spatio-temporal baseline, U-BARN computes temporal attention at a full spatial resolution. In this study, we demonstrate that the designed spatio-temporal architecture of the U-BARN is relevant as it slightly outperforms the U-TAE on a crop classification task.
Additionally, we introduce a BERT-inspired pretext task for pre-training U-BARN to reconstruct masked patch from a patch time series. We then assess the quality of the learned feature by studying two ways of using the pre-trained U-BARN weights: either frozen or fine-tuned. First, we demonstrate that the frozen and pre-trained U-BARN representation contains meaningful information for crop classification. Additionally, the fine-tuned U-BARN^{FT} significantly outperforms both U-TAE and non-pre-trained U-BARN^{e2e} when the number of labeled samples is low. However, the gain in classification performance decreases with an increase in labeled samples. Eventually, our results also indicate that the percentage of patches masked during the pre-training task has a significant impact on the classification performance. With our pre-training task, we suggest using a masking rate of 60% with U-BARN.

Although our results are promising, we believe that the current pre-training task does not adequately incorporate spatial features. Therefore, developing a spatial self-supervised strategy, may be a promising direction to improve classification performance. Additionally, the temporal dimension of the learned representation is the same as the input time series. In the case of irregularly sampled time series, the classifier in the downstream task need to be able to manage this kind of data. Moreover, the usual solutions (interpolation, gap-filling, or temporal reduction) may lead to a loss of information. To address this limitation, we suggest altering the network to achieve a fixed temporal sampling. Besides, a latent space with

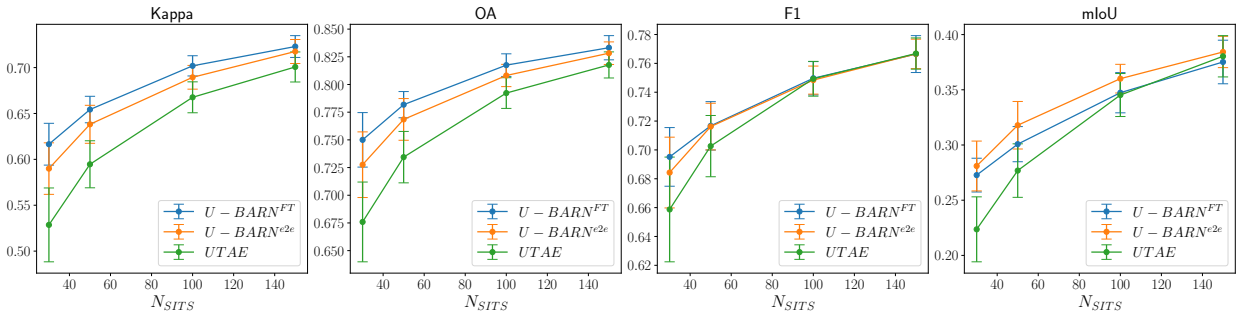

Figure 11. Evolution of the Kappa, OA, F1, and mIoU scores as a function of the number of SITS in the training data-set PASTIS for different SITS classifiers: U-BARN^{FT}-SC, U-BARN^{e2e}-SC and UTAE

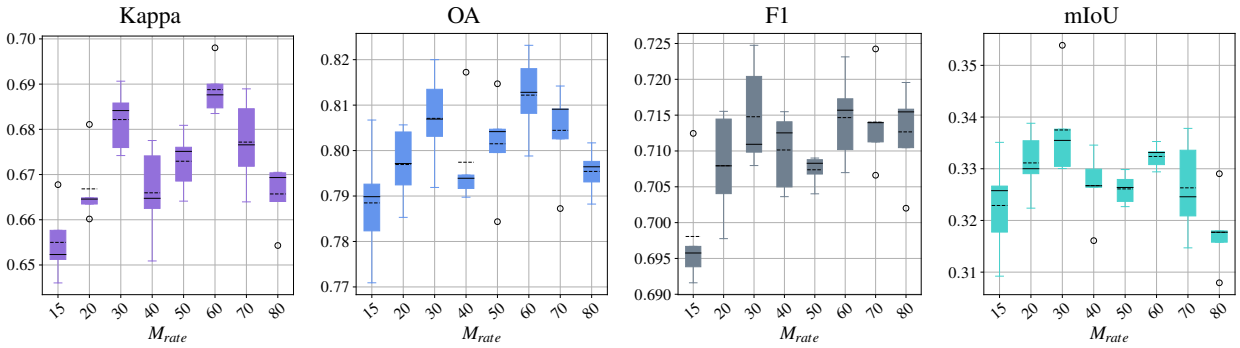

Figure 12. Evolution of the classification performances of U-BARN^{FR}-SC on PASTIS data-set for different masking rate in the pre-training task.

fixed-dimension is easier to analyze and interpret. Finally, we plan to apply this architecture to other downstream tasks and extend our self-supervised scheme to multi-modal data.

References

[1] G. Giuliani, G. Camara, B. Killough, and S. Minchin, "Earth observation open science: Enhancing reproducible science using data cubes," *Data*, vol. 4, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2306-5729/4/4/147

[2] F. Petitjean, J. Inglada, and P. Gancarski, "Satellite image time series analysis under time warping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 8, pp. 3081–3095, 2012. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2011.2179050

[3] D. R. Panuju, D. J. Paull, and A. L. Griffin, "Change detection techniques based on multispectral images for investigating land cover dynamics," *Remote Sensing*, vol. 12, no. 11, p. 1781, 2020. [Online]. Available: http://dx.doi.org/10.3390/rs12111781

[4] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2020.12.010

[5] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: a review," *Remote Sensing*, vol. 12, no. 15, p. 2495, 2020. [Online]. Available: http://dx.doi.org/10.3390/rs12152495

[6] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022.

[7] P. Berg, M.-T. Pham, and N. Courty, "Self-supervised learning for scene classification in remote sensing: Current state of the art and perspectives," *Remote Sensing*, vol. 14, no. 16, p. 3995, 2022. [Online]. Available: http://dx.doi.org/10.3390/rs14163995

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] A. Saeed, V. Ungureanu, and B. Gfeller, "Sense and learn: Self-supervision for omnipresent sensors," *Machine Learning with Applications*, vol. 6, p. 100152, 2021. [Online]. Available: http://dx.doi.org/10.1016/j.mlwa.2021.100152

[11] H. Li, Y. Li, G. Zhang, R. Liu, H. Huang, Q. Zhu, and C. Tao, "Global and local contrastive self-supervised learning for semantic segmentation of hr remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2022.3147513

[12] M. Hu, C. Wu, and L. Zhang, "Hypernet: Self-supervised hyperspectral spatial-spectral feature understanding network for hyperspectral change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2022.3218795

[13] C. Liu, H. Sun, Y. Xu, and G. Kuang, "Multi-source remote sensing pretraining based on contrastive self-supervised learning," *Remote Sensing*, vol. 14, no. 18, p. 4632, 2022. [Online]. Available: http://dx.doi.org/10.3390/rs14184632

[14] Y. Yuan, L. Lin, Z.-G. Zhou, H. Jiang, and Q. Liu, "Bridging optical and sar satellite image time series via contrastive feature extraction for crop classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 222–232, 2023. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2022.11.020

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, 10 2018. [Online]. Available: https://arxiv.org/abs/1810.04805v2

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 979–15 988.

[17] Y. Yuan and L. Lin, "Self-supervised pretraining 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. [Online]. Available: http://dx.doi.org/10.1109/JSTARS.2020.3036602

[18] Y. Yuan, L. Lin, Q. Liu, R. Hang, and Z.-G. Zhou, "Sits-former: A pre-trained spatio-spectral-temporal representation model for sentinel-2 time series classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102651, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243421003585

[19] V. S. F. Garnot and L. Landrieu, "Panoptic segmentation of satellite image time series with convolutional temporal attention networks," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10 2021, pp. 4852–4861. [Online]. Available: http://dx.doi.org/10.1109/ICCV48922.2021.00483

[20] C. Pelletier, G. 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. [Online]. Available: http://dx.doi.org/10.3390/rs11050523

[21] Z. Sun, L. Di, and H. Fang, "Using long short-term memory recurrent neural network in land cover classification on landsat and cropland data layer time series," *International Journal of Remote Sensing*, vol. 40, no. 2, pp. 593–614, 2018. [Online]. Available: http://dx.doi.org/10.1080/01431161.2018.1516313

[22] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel, "Land cover classification via multitemporal spatial data by deep recurrent neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1685–1689, 2017. [Online]. Available: http://dx.doi.org/10.1109/LGRS.2017.2728698

[23] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, p. 129, 2018. [Online]. Available: http://dx.doi.org/10.3390/ijgi7040129

[24] D. H. T. Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, and P. Maurel, "Deep recurrent neural networks for winter vegetation quality mapping via multitemporal sar sentinel-1," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 464–468, 2018. [Online]. Available: http://dx.doi.org/10.1109/LGRS.2018.2794581

[25] E. Ndikumana, D. H. T. Minh, N. Baghdadi, D. Courault, and L. Hossard, "Deep recurrent neural network for agricultural classification using multitemporal sar sentinel-1 for camargue, france," *Remote Sensing*, vol. 10, no. 8, p. 1217, 2018. [Online]. Available: http://dx.doi.org/10.3390/rs10081217

[26] C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu, "Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas," *Remote Sensing of Environment*, vol. 187, pp. 156–168, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425716303820

[27] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. G. Pensa, and S. Dupuy, "m^3Fusion: a deep learning architecture for multiscale multimodal multitemporal satellite data fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4939–4949, 2018. [Online]. Available: http://dx.doi.org/10.1109/JSTARS.2018.2876357

[28] R. Interdonato, D. Ienco, R. Gaetano, and K. Ose, "Duplo: a dual view point deep learning architecture for time series classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 149, pp. 91–104, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2019.01.011

[29] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924–935, 2019. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2018.2863224

[30] S. Mohammadi, M. Belgiu, and A. Stein, "3d fully convolutional neural networks with intersection over union loss for crop mapping from multi-temporal satellite images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 5834–5837.

[31] S. Ji, C. Zhang, A. Xu, Y. Shi, and Y. Duan, "3d convolutional neural networks for crop classification with multi-temporal remote sensing images," *Remote Sensing*, vol. 10, no. 2, p. 75, 2018. [Online]. Available: http://dx.doi.org/10.3390/rs10010075

[32] M. Rußwurm and M. Körner, "Self-attention for raw optical satellite time series classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 421–435, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2020.06.006

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[34] V. Sainte Fare Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention," *CVPR*, 2020.

[35] V. S. F. Garnot and L. Landrieu, *Lightweight Temporal Self-attention for Classifying Satellite Images Time Series*, ser. Advanced Analytics and Learning on Temporal Data. Springer International Publishing, 2020, pp. 171–181. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-65742-0_12

[36] W. Zhang, H. Zhang, Z. Zhao, P. Tang, and Z. Zhang, "Attention to both global and local features: a novel temporal encoder for satellite image time series classification," *Remote Sensing*, vol. 15, no. 3, p. 618, 2023. [Online]. Available: http://dx.doi.org/10.3390/rs15030618

[37] M. Rußwurm and M. Körner, "Convolutional lstms for cloud-robust segmentation of remote sensing imagery," *arXiv preprint arXiv:1811.02471*, 2018.

[38] V. Sainte Fare Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Time-Space Tradeoff in Deep Learning Models for Crop Classification on Satellite Multi-Spectral Image Time Series," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. Yokohama, Japan: IEEE, Jul. 2019, pp. 6247–6250. [Online]. Available: https://hal.science/hal-02386701

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[40] I. Misra, C. L. Zitnick, and M. Hebert, *Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification*, ser. Computer Vision ECCV 2016. Springer International Publishing, 2016, pp. 527–544. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_32

[41] A. Mohamed, H. yi Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaloe, T. N. Sainath, and S. Watanabe, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2022.3207050

APPENDIX A
DETAILED U-BARN ARCHITECTURE

Table V
HYPER-PARAMETER OF THE ARCHITECTURE OF THE UNET ENCODER. DOWNBLOCK ARCHITECTURE IS DETAILED IN FIG. 13

Block Name	Input dimension	Output dimensions
Input Convolution	(B*T,64,64,10)	(B*T,64,64,64)
Down Block 1	(B*T,64,64,64)	(B*T,64,64,64)
Down Block 2	(B*T,64,64,64)	(B*T,64,64,64)
Down Block 3	(B*T,64,64,64)	(B*T,64,64,128)

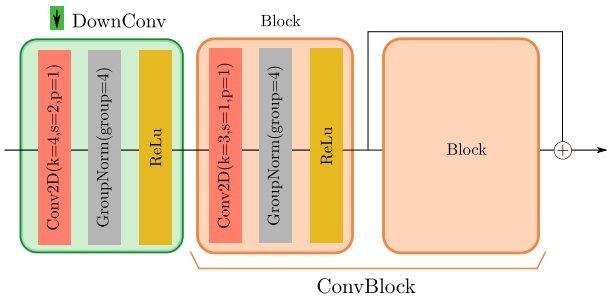

Figure 13. Down Block description

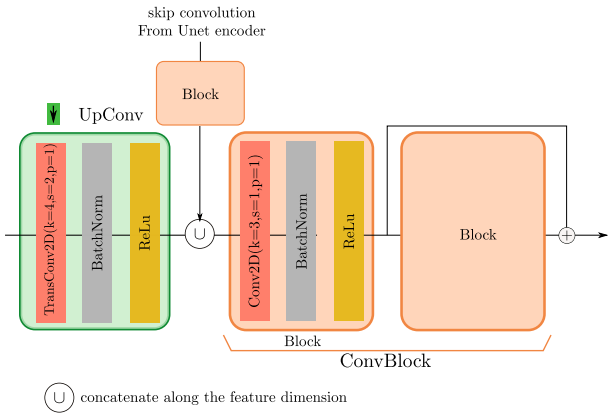

Figure 14. Up Block description

Table VI
ARCHITECTURAL HYPER-PARAMETERS OF THE TRANSFORMER

N_{layers}	N_{head}	$attn_{dropout}$	dropout	d_{model}	d_{hidden}
3	4	0.1	0.1	64	128

APPENDIX B
GENERATION OF SMALL LABELLED DATA-SET FROM PASTIS

A probability p_{P_i} Eq. 5 to draw the patch is computed on each patch. This probability increases with the number of pixels belonging to scarce classes in the patch. More precisely, the following protocol is established:

1) A score s_k, is computed . $s_k = \alpha \times \frac{1}{n_k}$ is inversely proportional to the total number n_k of pixels from the class k in the selected training data-set, α is a constant normalization so $\alpha \Sigma_k s_k = 1$.
2) For each patch P_i, the sum of the number of elements in the patch ($n_k^{P_i}$) from the class k, is weighted by the previously computed class probability s_k. The resulting score is then normalized by the total number of pixels belonging to the K classes in the patch. Eventually, the constant Λ is used, so the sum of p_{P_i} equals to 1.

$$p_{P_i} = \frac{\Sigma_k n_k^{P_i} * s_k}{\Sigma_k n_k^{P_i}} \times \Lambda \tag{5}$$

3) For each patch, we attribute disjoint interval contained in [0,1), of length equal to the patch probability
4) We draw N_{SITS} random numbers between [0,1). The patches which contains these random numbers constitute this tiny training data-set.