Self-Supervised Spatio-Temporal Representation Learning of Satellite Image Time Series

Iris Dumeur, *Student Member, IEEE*, Silvia Valero, *Member, IEEE*, and Jordi Inglada

***Abstract*—In this article, a new self-supervised strategy for learning meaningful representations of complex optical satellite image time series (SITS) is presented. The methodology proposed, named Unet-BERT spAtio-temporal Representation eNcoder (U-BARN), 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 but adapted to SITS is proposed. A Sentinel-2 large-scale unlabeled dataset is used to pretrain U-BARN. During the pretraining, U-BARN processes annual time series composed of a maximum of 100 dates. 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 crop dataset (PASTIS) as well as a dense land cover dataset (MultiSenGE). Two ways of exploiting U-BARN pretraining are considered: either U-BARN weights are frozen or fine-tuned. The obtained results demonstrate that representations of SITS given by the frozen U-BARN are more efficient for land cover and crop classification than those of a supervised-trained linear layer. Then, we observe that fine-tuning boosts U-BARN performances on MultiSenGE dataset. In addition, we observe on PASTIS, in scenarios with scarce reference data that the fine-tuning brings a significative performance gain compared to fully supervised approaches. We also investigate the influence of the percentage of elements masked during pretraining on the quality of the SITS representation. Eventually, semantic segmentation performances show that the fully supervised U-BARN architecture reaches better performances than the spatio-temporal baseline (U-TAE) on both downstream tasks: crop and dense land cover segmentation.**

***Index Terms*—Representation learning, satellite image time series (SITS), self-supervised learning (SSL), spatio-temporal network, transformer, Unet.**

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 multispectral 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 nonstationary, multivariate, 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 datasets [6]. This strategy enables the pretraining of deep models on large unlabeled datasets for later fine-tuning a shallow network on a downstream task. Therefore, self-supervised pretraining methods can be a solution for applications collecting small labeled datasets, where deep models cannot be trained from scratch.

Recent reviews [6], [7] have highlighted the great opportunities of SSL 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 autoencoders [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 SSL 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 multispectral 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, Liu et al. [13] processed pairs of single S1 and S2 images, while Yuan et al. [14] handled pairs of S1, S2 SITS. However, it should be noted that this latter contrastive approach on SITS to pretrain deep architectures is not unsupervised, as classification labels are utilized to generate

Manuscript received 27 September 2023; revised 22 November 2023 and 19 January 2024; accepted 19 January 2024. Date of publication 25 January 2024; date of current version 12 February 2024. This work was supported by the DeepChange Project under Grant ANR-DeepChange CE23. *(Corresponding author: Iris Dumeur.)*

The authors are with the CESBIO, Université de Toulouse, 31000 Toulouse, France (e-mail: iris.dumeur@univ-tlse3.fr; silvia.valero@cesbio.cnes.fr; jordi.inglada@cesbio.eu).

Digital Object Identifier 10.1109/JSTARS.2024.3358066

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 the features and information of the data itself. For instance, BERT [15] aims to recover masked words, and masked autoencoders (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 article presents a novel SSL method for capturing meaningful representations of complex optical SITS. The proposed methodology, named U-BARN, proposes an SSL strategy to learn a Unet-BERT spAtio-temporal Representation eNcoder. The first important contribution is the design of a new DL architecture that captures the spatio-temporal information contained in irregularly sampled multivariate 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 convolutional neural network (CNN) as in the SITS-Former [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. Our pretraining approach differs from SOTA SSL strategy on SITS [18], on significative points. First, the masking strategy is different, and the effect of the masking rate is studied. Then, in opposition to [18], our pretraining dataset contains cloudy images. Therefore, U-BARN can be applied on "real-world" downstream tasks where no cloud masks are available. Eventually, our unlabeled and labeled datasets have a great geographical variability, which demonstrate the scalability of our approach. 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 Section III-A. On the right, the use of pretrained U-BARN in the semantic segmentation downstream task is illustrated. In a nutshell, the main contributions of this article are as follows:

1) the construction of a novel spatio-temporal architecture for SITS, named U-BARN;
2) the self-supervised training of U-BARN with a generative pretext task;
3) the assessment of the self-supervised training strategy on two different downstream tasks.

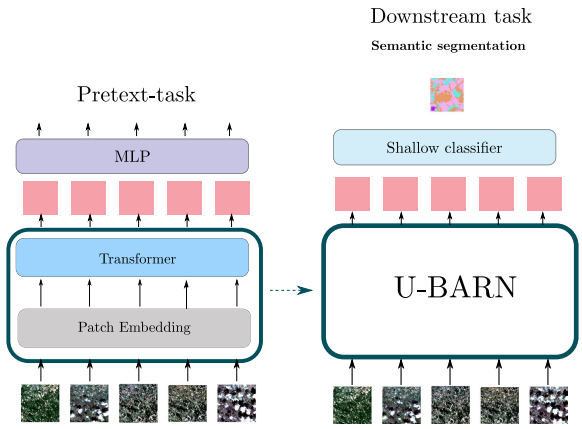

Fig. 1. Left: description of the proposed SSL strategy using BERT. Right: description of how representations are used for the downstream semantic segmentation task.

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 tasks defined by the labeled PASTIS dataset [19] and MultiSengGE [20]. First, the pretrained 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. In addition, different experiments are carried out to study the influence of the complexity of the pretraining task on the quality of the spatio-temporal representations. Lastly, a study of the U-BARN computational efficiency is conducted. The rest of this article is organized as follows.

1) A presentation of current state-of-the-art spatio-temporal architectures for SITS and existing SSL strategies are presented in Section II.
2) A detailed description of our methodology is given in Section III.
3) An explanation of the experimental setup is detailed in Section IV.
4) The results obtained from the different experiments are presented in Section V.
5) Finally, Section VI concludes this article.

For reproducibility, the large unlabeled S2 L2A dataset used to pretrain U-BARN [21] as well as the code[1] are available.

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 multitemporal data provide the most essential information to characterize land cover classes.

[1] [Online]. Available: https://src.koda.cnrs.fr/iris.dumeur/ssl_ubarn

For this reason, the earlier DL architectures exploiting recent SITS have not considered the spatial dimension of the data. For instance, TempCNN [22], which applies convolution on the temporal dimension, or recurrent neural networks (RNN) [23], [24], [25], [26], [27], which retain past timestamps information in memory, have been proposed. Although these architectures can outperform traditional approaches, such as random forest [28], existing literature [29], [30], [31], 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. 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 [31], 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 [29] and [30]. 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 fully connected (FC) network to predict the final class. In M^3-fusion [29], the architecture proposes the fusion of S2 pixel time series with Spot 6/7 very high spatial resolution patch images centered on the pixel of the time series. Features from temporal data are extracted by applying an 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 [30], 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 processes 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 [29], [30], [31] 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 3-D CNN to learn the local temporal features along with the spatial ones [32], [33]. These latter architectures process inputs with wider spatial dimensions, and in particular [32] fully convolutional architecture is efficient for segmentation map prediction. However, only short-temporal dynamics of the time series are learned by such architectures.

In addition, the use of the aforementioned temporal architectures on SITS suffer from important weaknesses. First, TempCNN do not handle irregularly sampled time series, which implies that all SITS are first resampled to a common gap-free temporal grid. Second, with RNN and TempCNN 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 [34] propose to apply the transformer network [35] 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 [34] compared to other architectures, such as Duplo [30] and TempCNN [22]. The study in [34] 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], [36], [37], [38], and spatio-temporal features [18], [19]. First, temporal approaches as [36], [37] propose different solutions to reduce the high computational complexity of the classical transformer network [35]. Both spectro-temporal models simplify the architecture by reducing the number of operations required to compute the attention score. The modified transformer, a temporal attention encoder (TAE), described in [36], 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 [36] is proposed in the lightweight temporal attention encoder (L-TAE) [37], where the master query is set as a network parameter. This last architecture outperforms TempCNN [22], [34], as well as architectures with RNN, Conv-LSTM [39], and Conv-GRU [40]. As the altered attention mechanism focuses on global attention, Zhang et al. [38] proposed 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 [29], [30], [31], 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 [41] 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. Eventually, recently vision Transformers (ViT) have also been proposed to process the spatial information [42]. In remote sensing, the TSViT [43], a fully attentional

architecture has been applied to SITS for crop classification. The TSViT is composed of a temporal transformer followed by a spatial transformer. Notably, to perform spatial attention, each image of the SITS is divided into smaller patches. Thus, the computational cost of the spatial attention mechanism increases quadratically with the number of patches. Therefore, the spatial context processed by the TSViT [43] is strongly limited by the hardware capacity. Consequently, our proposed methodology, U-BARN, combines a Unet with a transformer to capture rich and wide spatial and temporal correlations. Importantly, unlike the ViT, the Unet computational complexity is not quadratically linked to the input size. Besides, in contrast to the U-TAE [19], 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. Eventually, the temporal information is processed by a vanilla transformer network [35]. Indeed, the altered temporal attention mechanisms suggested in recent temporal model for SITS (TAE [36] and L-TAE [37]) collapse the temporal dimension to 1. Processing SITS through these networks prevent the use of SSL tasks aiming the reconstruction of masked input data.

B. Using Self-Supervised Pretext Tasks for Temporal Data

Self-supervised pretraining 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 [44]. In contrast, methods based on generative pretext tasks have learned to regenerate the input time series [45] based on some limited view of the data. Note that generative pretext tasks differ from generative models, which learn implicit distributions that allow us to sample new data. The reconstruction of masked tokens (e.g., embedded words or subwords) 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 authors' knowledge, SITS-BERT [17] was the first self-supervised strategy exploiting SITS. This last study proposed to learn spectro-temporal features from S2 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 multitemporal 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 [30], SITS-BERT [17], and Conv-RNN [25]. 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. Second, distribution shift can significantly impact fine-tuning performance. In this context, distribution shift means that, at inference time, the data are 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, we suppose that this approach might not adequately address the distribution shift issue. Third, while Rußwurm and Körner [34] have 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 nonfiltered or imperfectly filtered cloud data. Eventually, recently, the pretraining of ViT [42], as MAE, have been applied to SITS through SAT-MAE [46]. In computer vision, ViT traditionally split images into smaller patches, and spatial attention is computed between the various embedded patches. For SITS, the authors suggest splitting SITS along the spatial and temporal dimension into 3-D cubes. Computing spatio-temporal attention between all these embedded cubes drastically increases the number of operations in the attention mechanism. In addition as SITS contain static objects (no spatial movement through time), computing spatio-temporal attention might not be worth the high computational cost. This spatio-temporal attention mechanism might prevent the processing long SITS. Indeed, while SAT-MAE shows interesting results on images, their temporal approach has solely been pretrained on RGB SITS composed of three dates.

III. Proposed Methodology

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

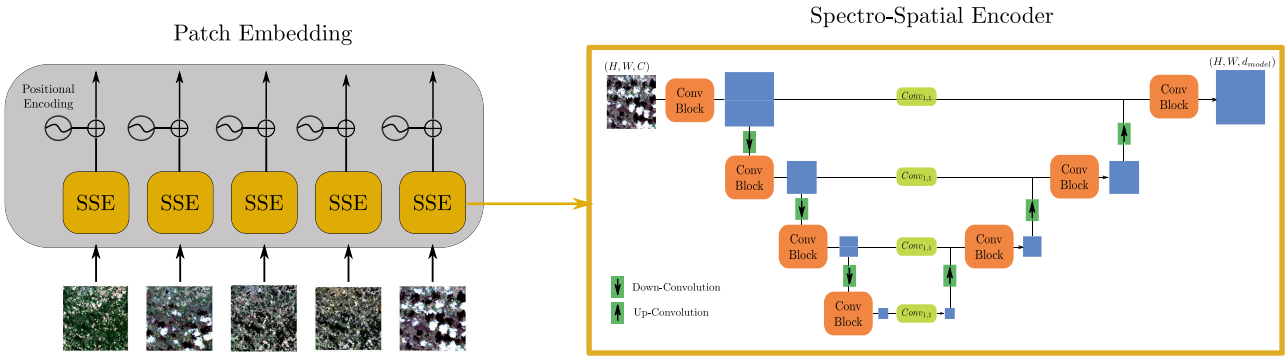

Fig. 2. Left: Overview of the patch embedding. A spatial-spectral encoder (SSE) embeds each patch into a (h, w, d_{model}) feature map. A positional encoding is added on the resulting feature maps. Right: Detailed description of the SSE architecture.

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 $(b, t, d_{\text{model}}, h, w)$ 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 spatial-spectral 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 time series on the learned SSE feature maps, the classical positional encoding [35] is added to each encoded patch of size d_{model}. As denoted by (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 [36], a scaling constant of a 1000 is considered

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

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

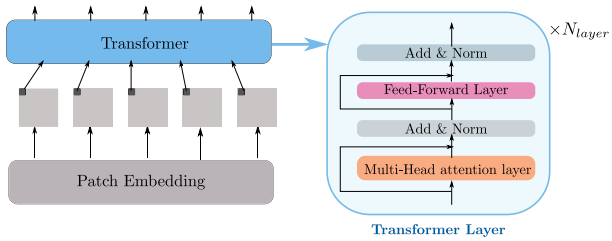

Fig. 3. Overall architecture of the spectro-temporal encoder. The transformer processes pixel-level time series.

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 $(b \times h \times w, t, d_{\text{model}})$. The backbone network is composed of multihead self-attention and feedforward layers, as detailed in Fig. 3.

The multihead 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 (2). These representations denoted by "query," $Q = W_Q X$ and "key" $K = W_K X \in \mathbb{R}^{t * d_{\text{model}}}$ are obtained by the learned projection matrices W_Q and W_K. As denoted by (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

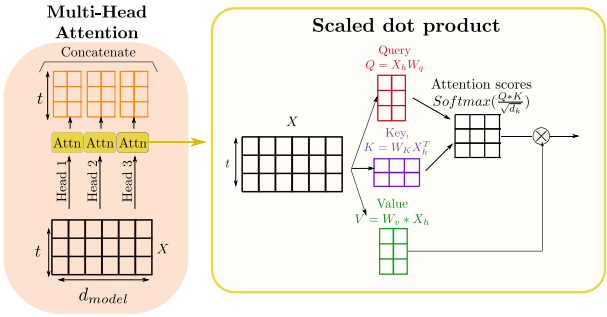

Fig. 4. Left: Description of the multiheadself-attention mechanism on a sequence of dimension (t, d_{model}). Right: Scaled-dot product on time series.

"value" $V = W_V X$. These weights give indication on which acquisitions are important for the training task

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right) V \qquad (2)$$

As demonstrated in [35], the computation of multihead scaled-attention products leads 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, scaled-dot product is computed in parallel on h set, of query, key, and value, called "heads," as depicted in Fig. 4. The resulting h time series are then concatenated and fed into feedforward layers that operate only on the feature (spectral) dimensions.

Feedforward layers are composed of two linear layers interspersed with an ReLu activation layer. Inside this feedforward block the first 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 4 heads each. The dimension of input and output features of the network are respectively set to $d_{\text{model}} = 64$ and $d_{\text{hidden}} = 128$. The architectural hyperparameters are detailed in Annex (see Appendix A).

B. Self-Supervised Strategy

Fig. 5 shows the overall framework of the proposed self-supervised pretraining 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. During this pretraining, the input time series are annual time series composed of a maximum of 100 dates. Specifically, the masking step is randomly applied on SSE output representations and a decoder multilayer perceptron (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 representations. Given an input time series, M_{rate} corresponds to the percentage of masked timestamps to be reconstructed. To

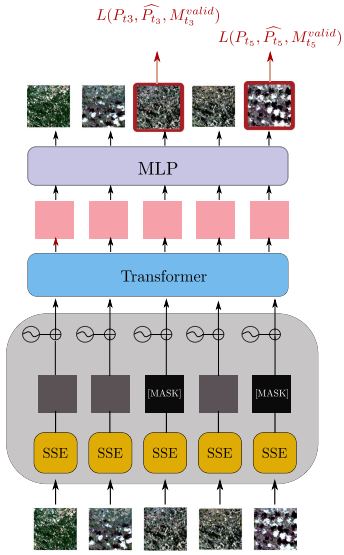

Fig. 5. Description of the proposed self-supervised strategy. A percentage M_{rate} of the feature maps encoded by the SEE are masked. U-BARN is trained to reconstruct the previously corrupted patch. The reconstruction loss is computed on the valid pixels, given by the binary mask M^{valid} associated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

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 mitigated in U-BARN masking strategy. To avoid disturbing the data distribution during the masking step, the proposed strategy consists in randomly permuting the spectro-spatial embedding values among selected encoded patches within a batch. Permuting pixels instead of masking them with a constant value, as 0, for instance, should have a lesser impact on the data mean and standard deviation, thus reducing the distribution shift. Specifically, an embedded pixel can be replaced by an embedded value from another date, another pixel location within the batch or another feature along the spectral dimension.

2) Decoder: The decoder, which is only used for pretraining 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 pretraining 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 $(b, t, d_{\text{model}}, h, w)$ 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. 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 $[P_{t_1}, \ldots, P_{t_{L_{\max}}}]$, a set T_S of masked dates, $n_{t_k}^{\text{valid}}$ the number of valid pixels in the patch P_{t_k}, the resulting loss can be expressed as (3). $\widetilde{P_{t_k}}$ is a patch that is corrupted at the SSE output by the previously described masking strategy. It is important to emphasize that the valid acquisition mask is solely utilized for the reconstruction loss in the SSL task. Consequently, validity masks are not required for downstream tasks

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

IV. Experimental Setup

First, the three S2 L2A datasets used in our different experiments are presented: the unlabeled large scale dataset used for pretraining U-BARN and the two downstream labeled datasets: PASTIS and MultiSenGE.

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

A. Datasets

S2 images processed to surface reflectances (L2A) by Theia are used in this study. For these datasets, 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 bicubic interpolation. A robust data normalization is applied on S2 L2A reflectances. First, the scaling technique of (4a) using the 0.05 and 0.95 quantiles is applied to remove data outliers by clipping the data. Second, the data are centered by subtracting the median value of x_{clip} to each spectral band and dividing the result by the dynamic data range [see (4b)].[2] Band statistics used to normalize the two S2 datasets are computed on the large unlabeled pretraining training dataset. Furthermore, in remote sensing due to memory limitations, deep neural networks usually do not process full satellite images. Therefore, smaller images denoted as "patches" are manipulated by UBARN. Specifically, our network processes patch time series of spatial dimension of (64 × 64). As the various datasets used might contain wider patches, a random crop transformation is operated during training. For validation and testing, the spatial crop is not random, and therefore, a center crop transform[3] is applied on the patch time series

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

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

1) Large-Scale Unlabeled Pretraining Dataset: The dataset is composed of 13 tiles acquired by S2 over France. The corresponding validity masks (noncorrupted pixels) are built by considering edge, saturation, and cloud information. Specifically, the cloud mask is built with MAJA [47]. 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 PASTIS dataset and the unlabeled dataset, as shown in Fig. 6. We have more diverse pretraining dataset, compared to that of the SITS-Former [18] dataset, which is only composed of SITS from 3 S2 tiles from 2018 or 2019. The U-BARN pretraining is performed by considering 9 different S2 tiles acquired from 2018 to 2020. In each of these tiles, 10 smaller regions of interest of size 1024 × 1024 are randomly selected. The disjoint validation dataset 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 hyperparameters. The validation dataset is used to select the best model weights, which are then used for the PASTIS downstream task. A more exhaustive description of the unlabeled dataset is given in Table I. Ultimately, the pretraining dataset is composed of annual SITS with dimensions ($t = 100$, $c = 10$, $h = 64$, $w = 64$). If the SITS have a temporal dimension lower than 100 dates, a temporal padding is applied.

2) PASTIS: This labeled S2 dataset 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

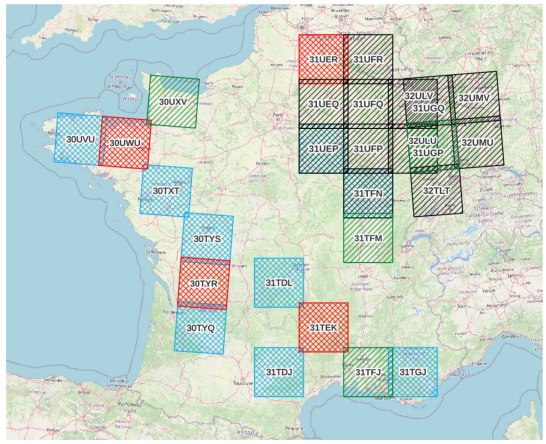

Fig. 6. Description of the S2 datasets used for pretext and downstream tasks. The unlabeled dataset for pretraining is composed of two disjoint datasets: training (tiles in blue) and validation (tiles in red). S2 tiles in the labeled datasets are shown in green and black, respectively, for PASTIS and MultiSenGE.

TABLE I
Pretrain Dataset Description

Dataset	S2 tiles	Year
Train	T30TXT, T30TYQ, T30TYS, T30UVU, T31TDJ, T31TDL, T31TFN, T31TGJ, T31UEP	2018–2020
Validation	T30TYR, T30UWU, T31TEK, T31UER	2016–2019

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

[3][Online]. Available: https://pytorch.org/vision/main/generated/torchvision.transforms.CenterCrop.html

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

TABLE III
DESCRIPTION OF THE LAND COVER CLASSES USED IN MULTISENGE [48]

Original level 1 typology	New level 3 typology
Urban areas	Dense built-up
	Sparse built-up
	Specialized built-up areas
	Specialized but vegetative areas
	Large scale network
Agricultural areas	Arable lands
	Vineyards
	Orchards
	Grasslands
	Groces, hedges
Forest and seminatural areas	Forest
	Open spaces, mineral
Wetlands	Wetlands
Water surfaces	Water surfaces

September 2018 to November 2019, only data from January 2019 to November 2019 is considered in our experiments. This requirement is imposed by our pretraining dataset, which is composed of annual time series. Furthermore, within this dataset, it has been estimated that 28% of images exhibit partial cloud cover according to [19], and no corresponding cloud masks are provided.

The complete dataset 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 dataset, 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).

3) MultiSenGE: MultiSenGE is a multitemporal dataset, which provides dense land cover labels over the Eastern region of France. We have used 8115 patches time series from 2020 with a spatial dimension of 256×256 pixels. Based on the LULC datastable named OCSGE2-GEOGRANDEST[4] and BDTOPO-IGN[5], this dataset is composed of 14 classes. As detailed in Table III, MultiSenGE is composed of 5 urban classes and 9 natural classes. In addition, to build this dataset exclusively images with a cloud cover less than 10% have been selected [48]. Similarly to PASTIS, no cloud masks are provided. In opposition to PASTIS dataset, MultiSenGE provides dense labels, therefore, all the pixels of a patch are classified. A random split is conducted to split the dataset between train (60%), validation(16%), and test (24%).

[4][Online]. Available: https://www.datagrandest.fr/portail/fr/tags/ocs-ge2

[5][Online]. Available: https://geoservices.ign.fr/documentation/donnees/vecteur/bdtopo

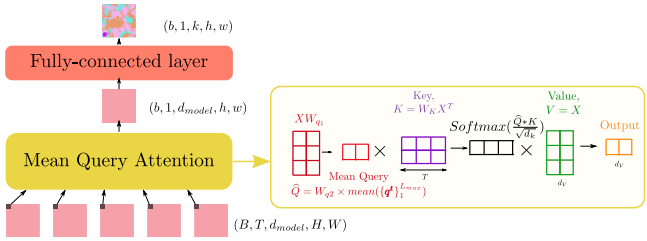

Fig. 7. Architecture of the SC and detailed description of the "mean-query" attention mechanism described in [36].

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. Second, since this is a segmentation task, the output of the SC should have no temporal dimension.

To meet both requirements, we have designed an SC, as shown in Fig. 7. To process inputs with different temporal dimension, the proposed SC uses the mean-query attention mechanism proposed in the TAE [36]. In this altered attention mechanism, a master query, which is the temporal average of the queries, is computed. In addition, in the computation of the "value" representation, the time series X is not projected by a matrix W_v, thus, $v = X$ in (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 an FC layer, to project the $(b, 1, d_{\text{model}}, H, W)$ 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.

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 cannot 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 pretrained 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 pretrained 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 pretrained U-BARN models, the previous self-supervised scenarios are compared with three training configurations supervised by the PASTIS dataset. The first one is denoted by U-BARN^{e2e} and corresponds to a trained end-to-end U-BARN encoder followed by the SC. When assessing with enough labeled data U-BARN^{e2e} encoder might be considered as the U-BARN^{FR} *higher bound* since frozen model performances may not surpass its fully supervised 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 pretrained 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 FC layer. For this situation, U-BARN is replaced by an FC layer, which operates exclusively on the feature (spectral) dimension. The FC layer increases the spectral dimension (10 spectral bands) to d_{model}. Then, the SC processes the SITS encoded by the FC layer. As the SC operates on the spectral and temporal dimension, this later configuration, denoted FC-SC, cannot capture spatial context.

Finally, the supervised spatio-temporal baseline U-TAE [19] is also considered in our experiments. We also have conducted a comparison with the TSViT architecture. Despite, the TSViT being positioned as a new spatio-temporal baseline in the PASTIS dataset, we have found that the U-TAE surpasses the TSViT in our experiments. These results are discussed in Appendix C, and we consider the U-TAE as the fully supervised spatio-temporal baseline 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 crop segmentation and MultiSenGE dense land cover segmentation downstream tasks. First, a qualitative evaluation of the pretext task training is proposed. Then, the quality of the representations learned by pretrained U-BARN models are evaluated by comparing the classification performances on both downstream tasks obtained by the aforementioned different training scenarios (see Section IV-C). The interest of using a pretrained U-BARN self-supervised encoder is corroborated by studying the robustness of the proposed methodology under reference data scarcity conditions. Afterward, the influence of the masking rate on the generalization capabilities of U-BARN representations is studied. Eventually, a computational efficiency study is conducted on U-BARN different configurations and U-TAE.

Each training (either pretraining 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.

TABLE IV
Classification Metrics Average and Standard Deviation Over PASTIS K-Folds for Different SITS Encoders

	Kappa	OA	F1	mIoU
FC-SC	0.738 ± 0.018	0.793 ± 0.015	0.509 ± 0.036	0.401 ± 0.028
U-BARN^{FR}	0.790 ± 0.011	0.832 ± 0.010	0.618 ± 0.017	0.501 ± 0.015
U-BARN^{FT}	**0.892** ± 0.011	**0.912** ± 0.009	**0.816** ± 0.018	**0.713** ± 0.022
U-BARN^{e2e}	**0.893** ± 0.010	**0.913** ± 0.008	**0.820** ± 0.013	**0.716** ± 0.017
U-TAE	**0.883** ± 0.012	**0.906** ± 0.009	0.803 ± 0.023	0.696 ± 0.027

The bold entities correspond to higher metrics values.

A. Qualitative Assessment of the Pretraining

This section presents an analysis of U-BARN's performance on the pretraining 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 Fig. 8). Therefore, we consider that U-BARN can successfully learn the temporal dynamic of the SITS during pretraining. 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 (3)]. Following [34], 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 pretraining is successful, as U-BARN is able to effectively solve the pretext-task.

B. Classification Performances on PASTIS and MultiSenGE Datasets

The classification performances on both labeled datasets obtained by the abovementioned described training scenarios are compared here. The two downstream tasks differ on two main points. First, in the MultiSenGE dataset has a dense semantic labeling, while in PASTIS all pixels, which do not belong to a known crop are not classified. Therefore, we assume that the spatial context should be better captured to successfully achieve the MultiSenGE labeling. However, we assume that to distinguish the 18 crops classes of PASTIS, compared to the 14 land cover classes, more complex temporal features are required. The U-BARN model is pretrained on the unlabeled dataset with the proposed generative pretext task strategy. The pretraining 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 OA. As we proceed to 5-fold training with PASTIS, mean and standard deviation of the classification metrics are given each time. For MultiSenGE downstream task, models are trained with two different seeds for each configuration. The overall results comparing the different training scenarios are reported in Table IV for PASTIS and

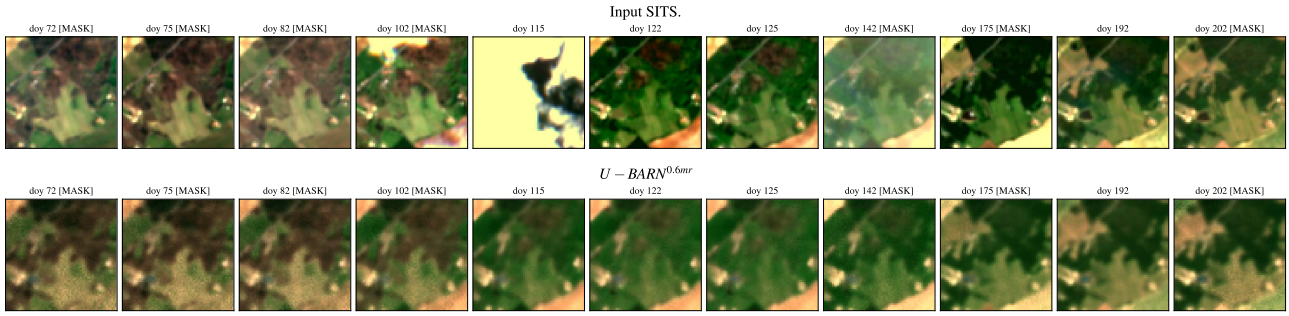

Fig. 8. Example of a patch (from the validation dataset) reconstruction achieved by U-BARN during pretraining. Only a part of the SITS is displayed. DOY of each patch are indicated. [MASK] indicates that the embedded patch was corrupted (see Section III-B1). The top row is the input SITS, and the bottom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

TABLE V
F1 SCORE PER CLASS ON PASTIS DATASET FOR DIFFERENT SITS ENCODERS

	FC-SC	U-BARN^{FR}	U-BARN^{FT}	U-BARN^{e2e}	U-TAE
Meadow	0.888 ± 0.011	0.904 ± 0.006	**0.945** ± 0.007	**0.945** ± 0.007	**0.939** ± 0.007
Soft winter wheat	0.850 ± 0.016	0.875 ± 0.009	**0.940** ± 0.008	**0.940** ± 0.010	**0.936** ± 0.008
Corn	0.903 ± 0.010	0.920 ± 0.007	**0.962** ± 0.005	**0.964** ± 0.004	**0.960** ± 0.006
Winter barley	0.628 ± 0.037	0.808 ± 0.037	**0.930** ± 0.018	**0.931** ± 0.017	**0.923** ± 0.019
Winter rapeseed	0.900 ± 0.017	0.901 ± 0.021	**0.963** ± 0.008	**0.962** ± 0.009	**0.961** ± 0.012
Spring barley	0.255 ± 0.058	0.626 ± 0.086	**0.779** ± 0.068	**0.788** ± 0.053	**0.768** ± 0.049
Sunflower	0.511 ± 0.052	0.647 ± 0.037	**0.871** ± 0.025	**0.862** ± 0.014	**0.860** ± 0.007
Grapevine	0.720 ± 0.033	0.790 ± 0.027	**0.916** ± 0.013	**0.916** ± 0.006	0.905 ± 0.016
Beet	0.855 ± 0.018	0.905 ± 0.016	**0.958** ± 0.027	**0.963** ± 0.019	**0.953** ± 0.026
Winter triticale	0.020 ± 0.012	0.211 ± 0.046	**0.677** ± 0.051	**0.688** ± 0.042	**0.683** ± 0.043
Winter durum wheat	0.605 ± 0.043	0.707 ± 0.027	**0.827** ± 0.026	**0.821** ± 0.032	0.798 ± 0.034
Fruits, vegetables, flowers	0.294 ± 0.094	0.409 ± 0.059	**0.721** ± 0.057	**0.727** ± 0.040	**0.697** ± 0.079
Potatoes	0.245 ± 0.175	0.533 ± 0.077	**0.748** ± 0.050	**0.734** ± 0.064	**0.704** ± 0.105
Leguminous fodder	0.332 ± 0.094	0.307 ± 0.051	**0.643** ± 0.060	**0.646** ± 0.062	**0.607** ± 0.057
Soybeans	0.706 ± 0.086	0.797 ± 0.012	0.937 ± 0.015	**0.948** ± 0.009	0.938 ± 0.013
Orchard	0.281 ± 0.055	0.580 ± 0.051	**0.775** ± 0.047	**0.782** ± 0.032	**0.761** ± 0.042
Mixed cereal	0.112 ± 0.063	0.079 ± 0.021	**0.545** ± 0.055	**0.563** ± 0.035	**0.528** ± 0.062
Sorghum	0.065 ± 0.070	0.123 ± 0.105	**0.558** ± 0.089	**0.569** ± 0.048	**0.541** ± 0.064

The bold entities correspond to higher metrics values.

TABLE VI
CLASSIFICATION METRICS OVER MULTISENGE FOR DIFFERENT SITS ENCODERS

	Kappa	OA	F1	mIoU
FC-SC	0.766 ± 0.001	0.838 ± 0.001	0.323 ± 0.001	0.254 ± 0.001
U-BARN^{FR}	0.772 ± 0.001	0.842 ± 0.001	0.356 ± 0.001	0.278 ± 0.002
U-BARN^{FT}	**0.855** ± 0.001	**0.898** ± 0.001	**0.506** ± 0.001	**0.421** ± 0.001
U-BARN^{e2e}	0.851 ± 0.001	0.895 ± 0.000	0.492 ± 0.003	0.407 ± 0.002
UTAE	0.832 ± 0.011	0.883 ± 0.007	0.426 ± 0.033	0.353 ± 0.030

Classification metrics are averaged along two trainings conducted with different seeds.
The bold entities correspond to higher metrics values.

Table VI for MultiSenGE. To bring detailed information on the classification of each class in the unbalanced datasets, the F1-score per class is also given in Tables V and VII. Eventually, on PASTIS dataset, the confusion matrix, from U-BARN^{FR} and FC-SC, are shown in Fig. 9 and example of the segmentation maps produced by the different networks is displayed in Fig. 10. Supplementary results over MultiSenGE dataset are available in Appendix D.

1) Frozen Encoder U-BARN^{FR}: As observed in Tables IV and VI, the performance of U-BARN^{FR} is intermediate between the FC-SC and the U-BARN^{e2e} on both downstream tasks. More precisely on PASTIS dataset, compared to the FC layer, the pretrained and frozen U-BARN^{FR} obtain a gain in Kappa of 0.052, 0.039 in OA, 0.109 in F1-score and 0.100 mIou. The F1-score per class also highlights that the classification gain differs for each class, with a significant improvement (at least 0.28 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, winter barley, sunflower, winter triticale, and fruit vegetables and flowers. The confusion matrices shown in Fig. 9 show that U-BARN^{FR} has fewer confusions than the FC-SC. For instance, U-BARN^{FR} performs better at distinguishing sunflower from potatoes or orchards from meadows. Compared to the FC layer encoding, U-BARN^{FR} also mitigates confusion

TABLE VII
F1 SCORE PER CLASS ON MULTISENGE DATASET

	FC − SC	U − BARN^{FR}	U − BARN^{FT}	U − BARN^{e2e}	UTAE
Dense built-up	0.008 ± 0.005	0.151 ± 0.001	**0.435** ± 0.030	0.365 ± 0.041	0.108 ± 0.080
Sparse built-up	0.597 ± 0.005	0.543 ± 0.010	**0.780** ± 0.003	0.770 ± 0.000	0.753 ± 0.004
Specialized built-up areas	0.244 ± 0.013	0.261 ± 0.034	**0.716** ± 0.002	0.566 ± 0.091	0.519 ± 0.103
Specialized but vegetative areas	0.000 ± 0.000	0.000 ± 0.000	**0.037** ± 0.011	0.017 ± 0.007	0.000 ± 0.000
Large scale networks	0.176 ± 0.068	0.199 ± 0.018	0.532 ± 0.006	**0.573** ± 0.032	0.402 ± 0.077
Arable lands	0.918 ± 0.002	0.921 ± 0.000	**0.956** ± 0.000	**0.957** ± 0.001	0.950 ± 0.002
Vineyards	0.336 ± 0.064	0.603 ± 0.029	**0.922** ± 0.004	**0.918** ± 0.002	0.877 ± 0.033
Orchards	0.000 ± 0.000	0.000 ± 0.000	0.055 ± 0.032	**0.077** ± 0.015	0.000 ± 0.000
Grasslands	0.723 ± 0.003	0.741 ± 0.001	**0.813** ± 0.003	**0.810** ± 0.001	0.791 ± 0.003
Groces, hegdes	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000
Forest	0.901 ± 0.001	0.903 ± 0.002	**0.934** ± 0.000	0.933 ± 0.001	0.923 ± 0.010
Open spaces, mineral	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000	**0.000** ± 0.000
Wetlands	0.000 ± 0.000	0.012 ± 0.015	**0.167** ± 0.005	**0.167** ± 0.037	0.027 ± 0.038
Water surfaces	0.622 ± 0.033	0.644 ± 0.000	**0.735** ± 0.017	**0.739** ± 0.019	0.619 ± 0.117

Classification metrics are averaged along two trainings conducted with different seed.
The bold entities correspond to higher metrics values.

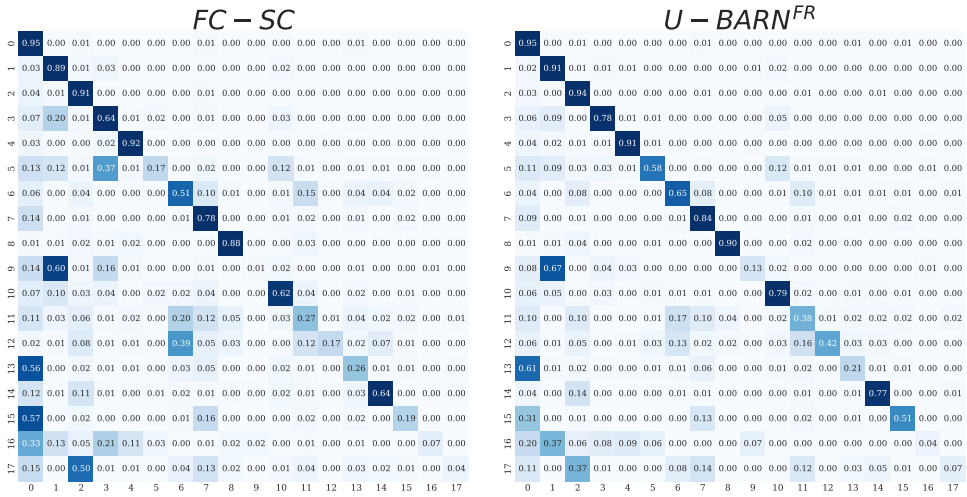

Fig. 9. Confusion matrices on the PASTIS segmentation task. On each confusion matrix, rows correspond to true label and columns to predictions. The matrices are normalized per row. The correspondence between PASTIS classes and the confusion matrix index is the following: {0: Meadow, 1: Soft winter wheat, 2: Corn, 3: Winter barley, 4: Winter rapeseed, 5: Spring barley, 6: Sunflower, 7: Grapevine, 8: Beet, 9: Winter triticale, 10: Winter durum wheat, 11: Fruits, vegetables, flowers, 12: Potatoes, 13: Leguminous fodder, 14: Soybeans, 15: Orchard, 16: Mixed cereal, 17: Sorghum}

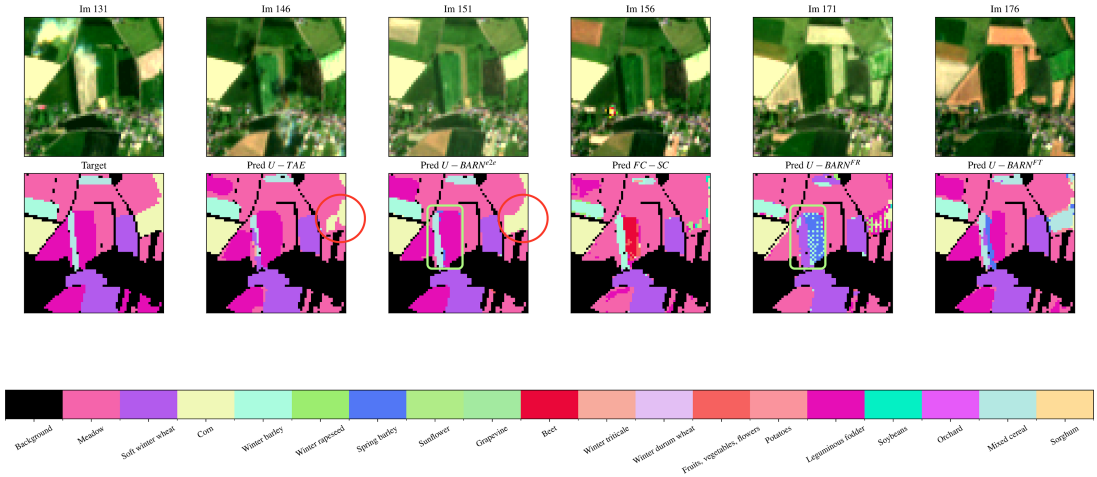

Fig. 10. Top row, some of the S2 RGB images which belong to input time series. Bottom row, different segmentation maps generated by the different networks. From left to right: target segmentation map, U-BARN^{e2e}, U-BARN^{FR}, U-BARN^{FT}, U-TAE, and the FC-SC predictions. The green boxes frame an area where U-BARN^{FR} predictions are spatially inconsistent compared to the fully supervised network U-BARN^{e2e}. The red circles highlight an area where U-TAE retrieves worse edges than U-BARN^{e2e}.

between spring and winter barley. Therefore, we conclude that the representations provided by U-BARN^{FR}, compared to SITS encoded by an FC layer, contain meaningful and discriminative information for the SCs. In addition, similar conclusions are found on MultiSenGE dataset, where U-BARN^{FR} outperforms FC-SC of 0.006 in Kappa, 0.004 in OA, 0.03 in F1 score, and 0.02 in mIoU. Since U-BARN^{FR} outperforms FC-SC on all classification metrics and on both downstream tasks, our self-supervised pretraining 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 us to completely learn the spatial correlations between pixels. As U-BARN^{e2e} segmentation maps do 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 fine-tuning configuration has two different behaviors depending on the downstream task. First, the global classification metrics on PASTIS presented in Table IV and the F1-score per class in Table V 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 the classification performance. We conjecture that the number and diversity of training labels available in the PASTIS dataset are sufficient to train the U-BARN^{e2e} model. This assumption is later investigated in Section V-C, where the classification performances of both U-BARN models are compared in scenarios with scarce reference data.

However, for the dense land cover segmentation task, fine-tuning seems to improve the performances. We could first suppose that the pretraining task is more adapted to learn features suitable for land cover classification. Another possibility is that MultiSenGE dataset does not have enough data to solve this complex dense land cover classification task. Therefore, pretraining the U-BARN on a large and diverse unlabeled dataset might help to extract meaningful spatio-temporal features.

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 on PASTIS dataset in Tables IV and V reveal close performances for both models. U-BARN^{e2e} has a significantly higher F1 score and mIoU. 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 V. We observe that F1 score are higher for classes, which are the most represented within the PASTIS dataset, such as Meadow or Corn. Conversely, less represented classes as Potatoes and Sorghum exhibit lower F1 score. Nevertheless, there is no direct relationship between the class size and its F1 score. This can be attributed to the fact that some classes may be more distinguishable than others. 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 dataset, small crops labels are discarded and considered as background, resulting in no assessment of segmentation of small items. In addition, it must be noted that the metrics found differ from those found in the original UTAE study [19]. This can be explained by the fact that we use only a part of the test dataset: a centered crop of 64×64 instead of the whole 128×128 pixels. The results slightly differ on MultiSenGE dense segmentation task. As shown in Table VI, the U-BARN^{e2e} outperforms U-TAE on all classification metrics. Thus, for dense segmentation task, U-BARN^{e2e} attention at full spatial resolution might be more advantageous. We notice that the F1 score strongly varies between the MultiSenGE classes (see Table VII). The various networks struggle to correctly classify the smallest classes: dense built-up, specialized but vegetative areas, orchards, groces and hedges, open spaces, and mineral as well as wetlands. As a conclusion, the overall results show that training the U-BARN architecture by using an end-to-end supervised task has better performances than the U-TAE [19] on both downstream tasks. While the gain of performances is modest for crop classification, it is more pronounced on dense land cover segmentation.

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 pretrained SITS-Former and end-to-end trained models increases as the number of training labels decreases. Therefore, a similar experiment conducted on the PASTIS dataset 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 the training dataset. In this experiment, U-BARN^{FT} is pretrained with a masking rate of 60%. As previously mentioned, the PASTIS dataset 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 dataset exhibits a strong classimbalance. To ensure that all classes are present in the generated reduced training datasets, the random selection of the patch time series follows the specific protocol detailed in Appendix B. Due to the small size of the resulting dataset, we have generated five smaller training datasets, 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 pretrained model with a training dataset 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 dataset composed of 30 patch time series, U-BARN^{FT} has a significantly higher mIoU and Kappa than U-BARN^{e2e}. The fine-tuning is, therefore, effective to boost performance when training with a reduced number of labels. Besides, on the 4 classification metrics with N_{SITS} lower or equal

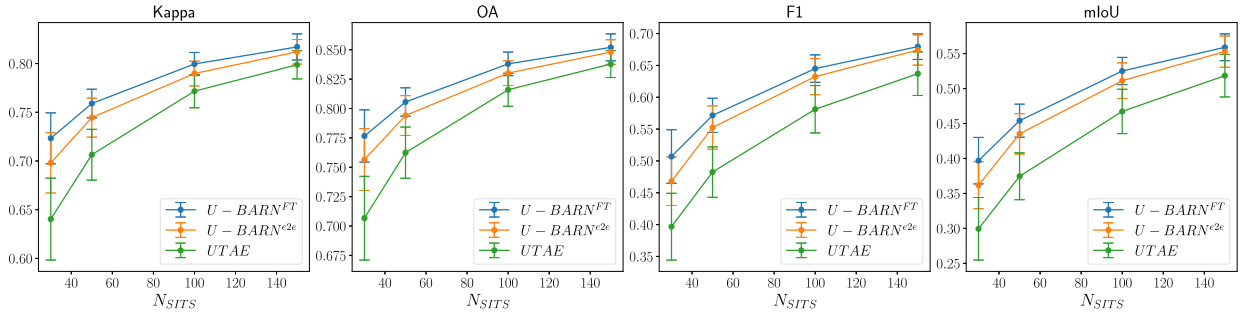

Fig. 11. Evolution of the Kappa, OA, F1, and mIoU scores as a function of the number of SITS in the training dataset PASTIS for different SITS classifiers: U-BARN^{FT}-SC, U-BARN^{e2e}-SC, and UTAE.

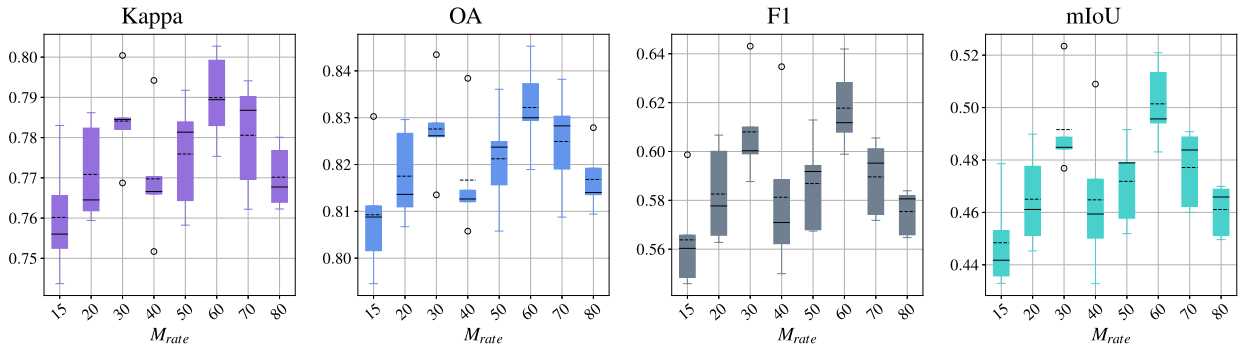

Fig. 12. Evolution of the classification performances of U-BARN^{FR}-SC on PASTIS dataset for different masking rate in the pretraining task.

to 100, U-BARN^{FT} and U-BARN^{e2e} outperform the U-TAE. We assume that because the U-TAE computes temporal attention at a low spatial resolution, the attention mechanism processes fewer pixel time series than the U-BARN and, therefore, is less competitive. On all the classification 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 pretraining, 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 pretraining task that can retrieve 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} pretrained 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 pretraining. As the masking rate increases, the number of "valid" dates used to reconstruct the corrupted patches diminishes, and the reconstruction loss during pretraining 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 pretraining our spatio-temporal architecture with SITS. In addition, results show that a masking ratio greater than 80% causes a significant drop in classification performances, indicating that the pretext-task might have become too difficult for training purposes.

TABLE VIII
COMPARISON OF U-BARN CONFIGURATIONS AND U-TAE WEIGHTS SIZE

Model name	Trainable weights	Total weights	Size in (MB)
U-TAE	1 086 969	1 086 969	4.35
U-BARN^{e2e}	1 122 323	1 122 323	4.49
FC-SC	14 547	14 547	0.06
U-BARN^{FR}	13 843	1 122 323	4.49
U-BARN^{FT}	1 122 323	1 122 323	4.49

E. Study of Computational Efficiency

The size of the various configurations as well as their training and inference times are compared in this section. First, Table VIII indicates, the number of trainable weights, the total number of weights, and the model size in MB. In addition, Table IX indicates the time of a training step, and an inference step. Time measures have been scaled by the FC-SC validation step time. Specifically, this table presents the median time to

TABLE IX
COMPARISON OF TRAINING AND VALIDATION TIME BETWEEN U-BARN CONFIGURATIONS AND U-TAE

Model name	Training step time	Validation step time
FC-SC	1.5	1.0
U-TAE	4.3	3.8
U-BARN^{FR}	10.5	10.2
U-BARN^{e2e}	11.5	10.2
U-BARN^{FT}	11.5	10.2

Time measures have been scaled by FC-SC validation step time.

process a random input of dimension (b, t, c, h, w), with $b = 2$, $t = 40$, $c = 10$, $h = 64$, $w = 64$ over 100 trials. These training and validation steps were executed on a single GPU Tesla V100. U-BARN^{e2e} is slightly bigger, in number of weights, than the U-TAE, as U-BARN has more transformer layers than the U-TAE and a different attention mechanism. However, U-BARN training and validation steps are 2,7 times slower than the U-TAE. Indeed, by computing attention at a low spatial resolution in the U-TAE, it drastically reduces the number of operations in the attention mechanism. Then, as expected, using the frozen configuration enables to drastically reduce the number of trainable weights, which decreases training time compared to U-BARN^{e2e} and U-BARN^{FT}.

VI. CONCLUSION

This article proposes a novel self-supervised methodology for learning spatio-temporal representations from SITS. The U-BARN architecture combines the strengths of Unet and transformer to extract informative and discriminative features from unlabeled datasets. We have assessed our network performances on two different segmentation scenarios: crop (PASTIS) and dense land cover (MultiSenGE). 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 outperforms the U-TAE on both downstream tasks. Although our architecture is less computationally efficient than the U-TAE, we have shown that this new design is more suitable to extract complex spatio-temporal features adapted for various tasks.

In addition, we introduce a BERT-inspired pretext task for pretraining U-BARN to reconstruct masked patches from annual patch time series, composed of a maximum of 100 dates. We present here a new way to corrupt the patches as well as investigate on the suitable masking rate. We then assess the quality of the learned feature by studying two ways of using the pretrained U-BARN weights: either frozen or fine-tuned. First, we demonstrate that the frozen and pretrained U-BARN representations contain meaningful information for crop and land cover classification. In addition, the fine-tuned U-BARN^{FT} significantly outperforms both U-TAE andnonpretrained U-BARN^{e2e} for dense land cover segmentation. On crop segmentation, U-BARN^{FT} exceeds both U-BARN^{e2e} and U-TAE performances 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 pretraining task has a significant impact on the classification performance. With our pretraining task, we suggest using a masking rate of 60% with U-BARN.

We are aware that compared to U-TAE, U-BARN is less computationally efficient. We assume that further research works should be pursued to reduce the number of operations of our architecture, while keeping temporal attention at a high-spatial resolution. Then, although our results are promising, further investigations should be conducted on MAE for SITS. Indeed, although we have stressed the importance of the masking rate value, we have not explored the influence of the masking value. Moreover, the use of asymmetric encoder–decoder architecture as proposed on [49], which avoids the use of [MASK] token in the encoder, should be explored. However, we believe that to extract complex spatio-temporal features, other pretraining tasks should also be studied to perform multitask pretraining. Given the important gap between our fully supervised configuration U-BARN^{e2e} and the frozen pretrained U-BARN^{FR}, we believe that masking solely on the temporal dimension is also not sufficient to extract complex spatio-temporal features. Specifically, we presume that the current pretraining task does not adequately incorporate spatial features. Therefore, combining the temporal masking strategy with a spatial self-supervised strategy, may be a promising direction to improve classification performances. In addition, 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 needs 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 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 multimodal data.

APPENDIX A
DETAILED U-BARN ARCHITECTURE

TABLE X
HYPERPARAMETERS OF THE ARCHITECTURE OF THE UNET ENCODER, WITH B AND T, RESPECTIVELY, THE BATCH AND TEMPORAL DIMENSIONS

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,32,32,64)
Down block 2	(B*T,32,32,64)	(B*T,16,16,64)
Down block 3	(B*T,16,16,64)	(B*T,8,8,128)

Downblock architecture is detailed in Fig. 13

APPENDIX B
GENERATION OF SMALL LABELED DATASET FROM PASTIS

TABLE XI
ARCHITECTURAL HYPERPARAMETERS OF THE TRANSFORMER

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

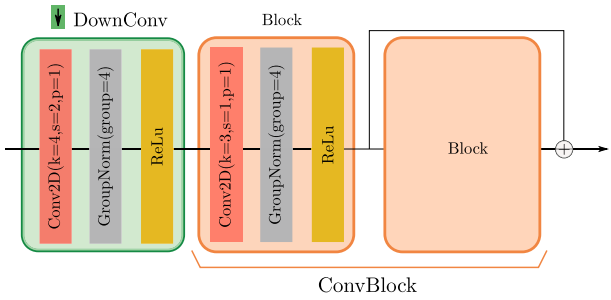

Fig. 13. Down Block description.

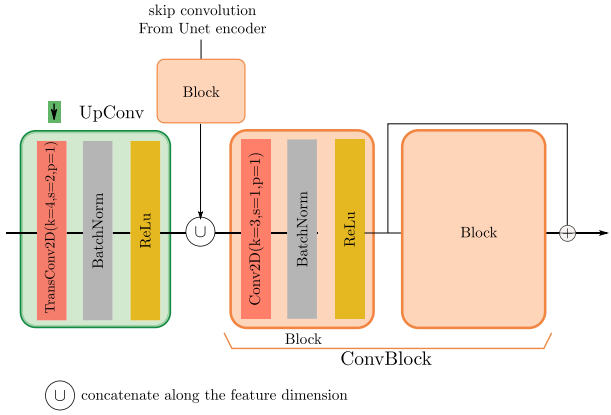

Fig. 14. Up block description.

TABLE XII
COMPARISON OF THE TSVIT MODEL SIZE COMPARED TO OTHER FULLY SUPERVISED ARCHITECTURES

Model Name	Trainable weights	Total weights	Size in (MB)
U-TAE	1086969	1086969	4.35
U-BARN^{e2e}	1122323	1122323	4.49
TSViT	1770116	1770116	7.08

TABLE XIII
COMPARISON OF THE TSVIT TRAINING AND VALIDATION TIME COMPARED TO THE OTHER FULLY SUPERVISED ARCHITECTURE COMPUTED ON ONE GPU TESLA V100

Model Name	Training step time	Validation step time
FC-SC	1.5	1.0
U-TAE	4.3	3.8
TSViT	10.2	9.4
U-BARN^{e2e}	11.5	10.2

A probability p_{P_i} to draw the patch is computed on each patch [see (5)]. 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 dataset, α is a normalization constant 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 that contain these random numbers constitute this tiny training dataset.

APPENDIX C
TSVIT PERFORMANCES ON PASTIS DATASET

The TSViT implementation, available at https://github.com/michaeltrs/DeepSatModels/tree/main, has been trained using the same training protocol as the other networks (U-TAE and U-BARN) on the PASTIS dataset. Table XIV shows the classification metrics on the PASTIS dataset while Tables XII and XIII compare the computational efficiency. First, according to Tables XII and XIII compared to the UTAE baseline, the TSViT is 1.6 times larger and its training step time 2.3 times slower. In Table XIV all networks underwent the same training procedure, except for TSViT$^{bs=4}$, which was trained with a batch dimension of 4. Surprisingly, we have found that the TSViT does not outperform the U-TAE on the PASTIS crop classification task. This discrepancy between our finding and [43] may be attributed to variations in the training protocol. First, unlike [43], we have pretrained TSViT on SITS with a spatial dimension of 64×64 rather than 24×24. Although, Tarasiou et al. [43] used small spatial dimensions due to computational constraints, it might also be a crucial factor, which affects the accuracy. We assume that a crop segmentation task might require a narrow spatial context. Therefore, employing a transformer that processes a long-range spatial correlation could be unnecessary and diminish the performances. In addition, the training hyperparameters (batch size, learning rate scheduler) differ between the original TSViT training and ours. Notably, TSViT seems to be highly sensitive to the batch dimension. As depicted in Table XIV TSViT$^{bs=4}$ significantly outperforms TSViT. In other words, increasing the batch size from 2 to 4 has a crucial impact on the classification performances. Furthermore, the two following points in the TSViT framework [43] are not clearly detailed and prevent us from fully understanding the TSViT results.

1) Tarasiou et al. [43] explained that their classification loss and metrics are computed while ignoring the background class. In opposition, the U-TAE [19] original paper omits the void class ("unknown crops") rather than the background class. However, despite this important label difference, the performance of the U-TAE presented in [43] is identical to its original paper [19]. This could potentially be a typographical error in this article, preventing us from fully understand the results.
2) The way SITS of different temporal length are processed remains unclear in [43]. Indeed, to create batch of SITS

TABLE XIV
CLASSIFICATION METRICS AVERAGE AND STANDARD DEVIATION OVER PASTIS K-FOLDS FOR DIFFERENT FULLY SUPERVISED ARCHITECTURE

	Kappa	OA	F1	IoU
U-BARN^{e2e}	**0.893** ± 0.010	**0.913** ± 0.008	**0.820** ± 0.013	**0.716** ± 0.017
U-TAE	**0.883** ± 0.012	**0.906** ± 0.009	0.803 ± 0.023	0.696 ± 0.027
TSViT$^{bs=4}$	0.869 ± 0.013	0.894 ± 0.011	0.785 ± 0.023	0.673 ± 0.027
TSViT	0.834 ± 0.025	0.866 ± 0.021	0.714 ± 0.046	0.595 ± 0.048

The bold entities correspond to higher metrics values.

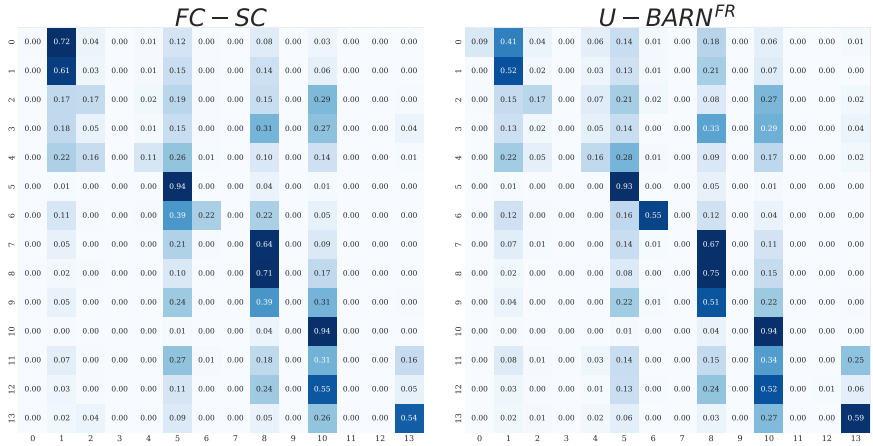

Fig. 15. Confusion matrices on the MultiSenGE segmentation task. On each confusion matrix, rows correspond to true label and columns to predictions. The matrices are normalized per row. The correspondence between MultiSenGE classes and the confusion matrix index is the following: {0: Dense built-up, 1: Sparse built-up, 2: Specialized built-up areas, 3: Specialized but vegetative areas, 4: Large scale network, 5: Arable lands, 6: Vineyards, 7: Orchards, 8: Grasslands, 9: Groces, Hedges, 10: Forest, 11: Open spaces, mineral 12: Wetlands, 13: Water surfaces}.

with different temporal length, SITS are padded along the temporal dimension. In the vanilla attention mechanism a "padding mask" is provided to the attention mechanism. Therefore, padded dates do not interfere in the attention mechanism. In their implementation, we have not found such masking in the attention mechanism.

APPENDIX D
MULTISENGE SUPPLEMENTARY RESULTS

See Fig. 15.

ACKNOWLEDGMENT

The authors would like to thank CNES for the provision of its high performance computing (HPC) [50] infrastructure to run the experiments presented in this article and the associated help. They also would like to thank Mathieu Fauvel and Julien Michel for their feedback on a preliminary version of this article.

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 Trans. Geosci. Remote Sens.*, vol. 50, no. 8, pp. 3081–3095, Aug. 2012, doi: 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 Sens.*, vol. 12, no. 11, 2020, Art. no. 1781, doi: 10.3390/rs12111781.

[4] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 173, pp. 24–49, 2021, doi: 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 Sens.*, vol. 12, no. 15, 2020, Art. no. 2495, doi: 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 Geosci. Remote Sens. Mag.*, vol. 10, no. 4, pp. 213–247, Dec. 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 Sens.*, vol. 14, no. 16, 2022, Art. no. 3995, doi: 10.3390/rs14163995.

[8] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[9] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.

[10] A. Saeed, V. Ungureanu, and B. Gfeller, "Sense and learn: Self-supervision for omnipresent sensors," *Mach. Learn. Appl.*, vol. 6, 2021, Art. no. 100152, doi: 10.1016/j.mlwa.2021.100152.

[11] H. Li et al., "Global and local contrastive self-supervised learning for semantic segmentation of HR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5618014, doi: 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 Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5543017, doi: 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 Sens.*, vol. 14, no. 18, 2022, Art. no. 4632, doi: 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 J. Photogrammetry Remote Sens.*, vol. 195, pp. 222–232, 2023, doi: 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," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 4171–4186. [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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15979–15988.

[17] Y. Yuan and L. Lin, "Self-supervised pretraining of transformers for satellite image time series classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 474–487, Nov. 2021, doi: 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," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 106, 2022, Art. no. 102651. [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 *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4852–4861, doi: 10.1109/ICCV48922.2021.00483.

[20] R. Wenger, A. Puissant, J. Weber, L. Idoumghar, and G. Forestier, "Multisenge: A multimodal and multitemporal benchmark dataset for land use/land cover remote sensing applications," *ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. V-3-2022, pp. 635–640, 2022. [Online]. Available: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-3-2022/635/2022/

[21] I. Dumeur, S. Valero, and J. Inglada, "Unlabeled sentinel 2 time series dataset: Self-supervised spatio-temporal representation learning of satellite image time series," 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7891924

[22] C. Pelletier, G. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 523, doi: 10.3390/rs11050523.

[23] 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," *Int. J. Remote Sens.*, vol. 40, no. 2, pp. 593–614, 2018, doi: 10.1080/01431161.2018.1516313.

[24] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel, "Land cover classification via multitemporal spatial data by deep recurrent neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1685–1689, Oct. 2017, doi: 10.1109/LGRS.2017.2728698.

[25] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS Int. J. Geo- Inf.*, vol. 7, no. 4, 2018, Art. no. 129, doi: 10.3390/ijgi7040129.

[26] D. H. T. Minh et al., "Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR sentinel-1," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 464–468, Mar. 2018, doi: 10.1109/LGRS.2018.2794581.

[27] 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 Sens.*, vol. 10, no. 8, 2018, Art. no. 1217, doi: 10.3390/rs10081217.

[28] 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 Sens. Environ.*, vol. 187, pp. 156–168, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425716303820

[29] 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 J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4939–4949, Dec. 2018, doi: 10.1109/JSTARS.2018.2876357.

[30] R. Interdonato, D. Ienco, R. Gaetano, and K. Ose, "Duplo: A dual view point deep learning architecture for time series classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 149, pp. 91–104, 2019, doi: 10.1016/j.isprsjprs.2019.01.011.

[31] 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 Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019, doi: 10.1109/TGRS.2018.2863224.

[32] 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 *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 5834–5837.

[33] 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 Sens.*, vol. 10, no. 2, 2018, Art. no. 75, doi: 10.3390/rs10010075.

[34] M. Rußwurm and M. Körner, "Self-attention for raw optical satellite time series classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 169, pp. 421–435, 2020, doi: 10.1016/j.isprsjprs.2020.06.006.

[35] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[36] V. Sainte FareL. GarnotS. Landrieu Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12322–12331.

[37] V. S. F. Garnot and L. Landrieu, *Lightweight Temporal Self-Attention for Classifying Satellite Images Time Series* (Advanced Analytics and Learning on Temporal Data). Berlin, Germany: Springer, 2020, pp. 171–181, doi: 10.1007/978-3-030-65742-0_12.

[38] 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 Sens.*, vol. 15, no. 3, 2023, Art. no. 618, doi: 10.3390/rs15030618.

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

[40] V. S. Fare, L. Garnot, S. L. Giordano, and N. Chehata, "Time-space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 6247–6250. [Online]. Available: https://hal.science/hal-02386701

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Interv.*, 2015, pp. 234–241.

[42] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[43] M. Tarasiou, E. Chavez, and S. Zafeiriou, "Vits for sits: Vision transformers for satellite image time series," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 10418–10428.

[44] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 527–544, doi: 10.1007/978-3-319-46448-0_32.

[45] A. Mohamed et al., "Self-supervised speech representation learning: A review," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1179–1210, Oct. 2022, doi: 10.1109/JSTSP.2022.3207050.

[46] Y. Cong et al., "SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022. [Online]. Available: https://openreview.net/forum?id=WBhqzpF6KYH

[47] L. Baetens, C. Desjardins, and O. Hagolle, "Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure," *Remote Sens.*, vol. 11, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/4/433

[48] R. Wenger, A. Puissant, J. Weber, L. Idoumghar, and G. Forestier, "Multisenge: A multimodal and multitemporal benchmark dataset for land use/land cover remote sensing applications," *ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. V-3-2022, pp. 635–640, 2022, [Online]. Available: http://dx.doi.org/10.5194/isprs-annals-V-3-2022-635-2022

[49] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[50] French spatial agency data processing centre, 2019.

Iris Dumeur (Student Member, IEEE) received the master's degree in mines civil engineering, majoring in computer science, from Ecole des Mines Nancy, Nancy, France, in 2021. She is currently working toward the Ph.D. degree with the Centre d'Etudes Spatiales de la Biosphère (CESBIO) Laboratory, Universite de Toulouse, Toulouse, France.

She is working on self-supervised learning strategies to generate meaningful representations of satellite image time series.

Silvia Valero (Member, IEEE) received the M.S. degree in electrical engineering from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2007, the M.S. degree in computer science from the Grenoble Institute of Technology (Grenoble-INP), Saint-Martin-d'Héres, France, in 2008, and the conjoint Ph.D. degree in signal processing from the Grenoble-INP and UPC in 2011.

She is currently an Associate Professor of Computer Science with the Université Paul Sabatier - IUT'A, Toulouse, France. Her research activity is currently carried out in the Centre d'Études Spatiales de la BIOsphere (CESBIO) Laboratory. Her research interests include image processing and machine learning for remote sensing data. Her current activities involve the exploitation of satellite image times series for land monitoring.

Jordi Inglada received the master's degree in telecommunications engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, and the École Nationale Supérieure des Télécommunications de Bretagne, Brest, France, in 1997, and the Ph.D. degree in signal processing and telecommunications from the Université de Rennes 1, Rennes, France, in 2000.

He is currently with the Centre National d'Etudes Spatiales (French Space Agency), Toulouse, France, where he is involved in the field of remote sensing image processing with the Centre d'Etudes Spatiales de la Biosphère (CESBIO) Laboratory, Toulouse, France. He is involved in the development of image processing algorithms for the operational exploitation of Earth observation images, mainly in the field of multitemporal image analysis for land use and cover change.