# End-to-end Learning for Land Cover Classification using Irregular and Unaligned SITS by Combining Attention-Based Interpolation with Sparse Variational Gaussian Processes

Valentine Bellet, Mathieu Fauvel, Jordi Inglada, Julien Michel

# End-to-end Learning for Land Cover Classification using Irregular and Unaligned SITS by Combining Attention-Based Interpolation with Sparse Variational Gaussian Processes

Valentine Bellet, *Graduate Student Member, IEEE,* Mathieu Fauvel, *Senior Member, IEEE,* Jordi Inglada, and Julien Michel

***Abstract*—In this article, we propose an approach using irregular and unaligned Sentinel-2 satellite image time series (SITS) for large-scale land cover pixel-based classification. We used end-to-end learning by combining an attention-based interpolator: the Multi-Time Attention Networks (mTAN), with a Sparse Variational Gaussian Processes (SVGP) classifier. The mTAN is used to project the irregular and unaligned SITS onto a fixed and reduced size representation. By using structured feature extraction, this representation is able to take into account the spectro-temporal structure of the SITS. Moreover, the spatial information is added to this representation by using the *spatial positional encoding*. This representation is given to the SVGP classifier and all the parameters are optimized using a loss function for classification. We ran experiments with irregular and unaligned Sentinel-2 SITS of the full year 2018 over an area of 200 000 km$^2$ (about 2 billion pixels) in the south of France. Using the representation from the mTAN instead of linearly interpolated SITS significantly improved the results in terms of classification accuracy of about 10 points for the overall accuracy. Moreover, the mTAN combined with the SVGP classifier is above the mTAN combined with Deep Learning classifiers (respectively seven and four points for the MLP and LTAE classifiers).**

***Index Terms*—Satellite Image Time-Series (SITS), Sentinel-2, Land Cover Map, Pixel-Based, Classification, Large Scale, Sparse Variational Gaussian Processes, Earth Observation (EO), Remote Sensing.**

## I. Introduction

IN March 2023, the final synthesis report of the Sixth Assessment Report (AR6) was released by the Intergovernmental Panel on Climate Change (IPCC). It is a comprehensive review of our knowledge of the climate change. Among the various findings, it reports that climate impacts on ecosystems are more intense and widespread than expected [1]. Among other things, they proposed to expand the use of digital technlogy for land use monitoring or sustainable land management which can help to reduce emissions from deforestation and land use change.

This work is supported by the Natural Intelligence Toulouse Institute (ANITI) from Universite Federale Toulouse Midi-Pyrenees under grant agreement ANITI ANR-19-PI3A-0004 (this PhD is co-founded by CS-Group and by the Centre National d'Etudes Spatiales (CNES)).

V. Bellet, M. Fauvel and J. Inglada are with CESBIO, Universite de Toulouse, CNES/CNRS/INRAe/IRD/UPS, 31000 Toulouse, France (e-mail: valentine.bellet@univ-toulouse.fr, mathieu.fauvel@inrae.fr, jordi.inglada@cesbio.eu)

Earth observation satellites provide a huge amount of raw data of different types (e.g. optical, radar). Extracting meaningful information from these raw EO data can help to monitor the Earth's surface changes and therefore can help to solve the challenges of climate change [2], [3]. For instance, the twin satellites Sentinel-2 provide free and open-access data with relevant features: high revisit time (every 5 days) and high spectral and spatial resolutions (four spectral bands at 10m, six at 20m and three at 60m per pixel) [4].

These optical Sentinel-2 satellite image time-series (SITS) that cover large continental surfaces with a short revisit cycle, bring the opportunity of large scale mapping. For example, land use or land cover (LULC) maps provide information about the physical and functional characteristics of the Earth's surface for a particular period of time. More precisely, land cover map usually refers to the physical land type (i.e. corn field or grassland) whereas land use map indicates how people are using the land (i.e. agriculture). To produce these LULC maps from the large quantity of SITS, automatic methods are required. In the last years, Machine Learning (ML) and then Deep Learning (DL) methods have shown very promising results in terms of performance accuracy [5]–[11]. Different methods were proposed for land cover classification using Sentinel-2 SITS at large scale. Recently, Bellet et al. [12] proposed a method based on Sparse Variational Gaussian Processes (SVGP). It provides similar classification performance to state-of-the-art methods such as conventional ML methods or DL methods. This method, like the others, requires data with a fixed-size.

Yet, Sentinel-2 SITS are irregularly sampled: observations are not equally spaced in time due to the presence of clouds or shadows. These time series are also unaligned: observations from two different satellite swaths have different temporal sampling grids. Even though some deep learning models such as Long short-term memory (LTSM) [13] can take into account irregular time series, they do not support unaligned time series. Usually, preprocessing techniques are used to transform these irregular and unaligned time series into time series that can be used by the classifier. In this context, Inglada et al. [7] proposed to linearly resample the observations onto a common set of latent dates. The obtained resampled observations from a full year were successfully used to produce land cover classification maps at country scale using SVGP [12].

However, information essential to the classification prediction can be lost when producing these resampled observations. Indeed, Li et al. [14] showed that an independent interpolation method directly followed by classification method performed worse than methods trained end-to-end. In this sense, Constantin et al. [15] did not use preprocessing techniques and proposed to jointly classify and reconstruct irregular pixel time series. Even if the reconstruction was good, the model did not compete with state of the art classifiers such as RF or Support Vector Machine (SVM). Using kernel-based interpolation, Lu et al. [16] produced a similarity function from two irregular time series. Few years later, Shukla et al. [17] also used a kernel smoother to form representation from irregular time series.

Authors of [18] proposed a method called Multi-Time Attention Networks (mTAN) which produces a fixed representation of an irregular and unaligned time series. It uses multiple continuous time embeddings coupled with attention mechanisms. The mTAN shows great performances for both interpolation and classification problems [18]. By using end-to-end training (mTAN coupled with a classifier), the performance results were similar to or better than state-of-the-art models.

In this paper, we propose to use end-to-end learning by combining the mTAN with the SVGP classifier. The mTAN is used to project onto a latent space of fixed and reduced size the irregular and unaligned time series. This representation is then given to the SVGP classifier and all the parameters are optimized using a loss function for classification. The SVGP classifier is described by a large number of parameters to be estimated highly depend on the spectro-temporal dimension of the data. Thus, the complexity of the model is proportional to this number of variables but also to the correlation between these variables. Using structured feature extraction in the mTAN module can be beneficial for the SVGP classifier.

Moreover, in large scale classification, due to different climatic and topographic conditions, there is a variation of the spectro-temporal signature over the spatial domain (i.e. non stationarity). By using spatial coordinates, with spatial stratification [7] or by learning a spatial-informed classifier [12], this non-stationarity can be taken into account and performances are improved. In this work, we propose to add the spatial information before the classification: in the latent representation obtained by the mTAN. The method used is based on *spatial positional encoding* defined in [19].

To sum up, Fig. 1 represents the end-to-end learning for the classification of one irregular and unaligned pixel time series $\mathbf{X}$. $\mathbf{Z}$ is its representation onto the fixed and reduced temporal grid with $\dim(\mathbf{Z}) < \dim(\mathbf{X})$. $h_{\boldsymbol{\theta}_1}$ and $f_{\boldsymbol{\theta}_2}$ represent respectively the mTAN and the SVGP classifier with their respective learnable parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. SVGP can be easily used in end-to-end learning as the gradient of the loss can be back-propagated. The loss $\mathcal{L}$ is used to optimized $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and to minimize the error between the predicted class $\hat{y}$ and the true class $y$. The loss is not modified from [12] and is based on Variational Inference.

The remainder of this paper is organized as follows. Section II describes how the mTAN is used to project irregular and unaligned pixel time series. Section III defines how the mTAN is adapted in order to fit to the classification task especially by using spectro-temporal reduction and spatial positional encoding. The experimental setup is detailed in Section IV. The associated results concerning the end-to-end training model (mTAN coupled with SVGP) are provided in Section V. Different competitive methods are studied and their associated results are presented in Section VI. Finally, Section VII concludes this paper and opens discussions on future works.

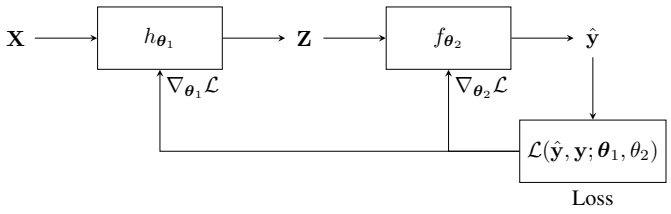



Fig. 1: End-to-end learning for the classification of one irregular and unaligned pixel time series $\mathbf{X}$.

## II. The irregular and unaligned pixel time series

This section describes how irregular and unaligned pixel time series is projected onto a fixed temporal grid in order to be used by the classifier. First, some notations and definitions which will be used throughout this paper are introduced. Then, the method at the basis of the proposed method (i.e. the mTAN) is presented and its modification is described in the last part.

### A. Notations and definitions

In this paper, the $i$th pixel time series $\mathbf{x}^i(t_k)$ at time $t_k$ is defined by its spectral measurements $\{x_1^i(t_k), ..., x_j^i(t_k), ..., x_D^i(t_k)\}$ with $i \in \{1, ..., N\}$, $N$ the number of pixels and $D$ the number of spectral features. Besides, two spatial coordinates $\psi_1^i$ and $\psi_2^i$ are associated to the pixel $\mathbf{x}^i$. Moreover, $y^i \in \{1, ..., C\}$ is the target value (i.e. the class membership) associated to the pixel $\mathbf{x}^i$, with $C$ the number of classes.

For a pixel $i$, a spectral feature $j$ is observed at $T_j^i$ timestamps: $\mathbf{T}_j^i = \{t_{j1}^i, ..., t_{jk}^i, ..., t_{jT_j^i}^i\}$, where $T_j^i$ is the number of valid observations (e.g., no clouds or shadows). As discussed in Section I, because of satellite swaths and weather we usually have unaligned time series, i.e., $\mathbf{T}_j^i \neq \mathbf{T}_j^{i'}$. In this work, we assume that all spectral features are available for each timestamp, i.e., $\mathbf{T}_j^i = \mathbf{T}_{j'}^i = \mathbf{T}^i$. This is commonly the case when working with only one sensor. As an illustration, Fig. 2 represents two real irregular and unaligned pixel time series acquired with Sentinel-2.

We defined the set of all timestamps $\mathbf{T}$ such as:

$$\begin{aligned}\mathbf{T} &= \bigcup_{i=1}^{N} \mathbf{T}^i \\ &= \{t_1, ..., t_k, ..., t_T\}\end{aligned}$$

with $T$ the total number of observations. For each pixel, we define a mask time series $\mathbf{m}^i \in \{0, 1\}^T$ such as

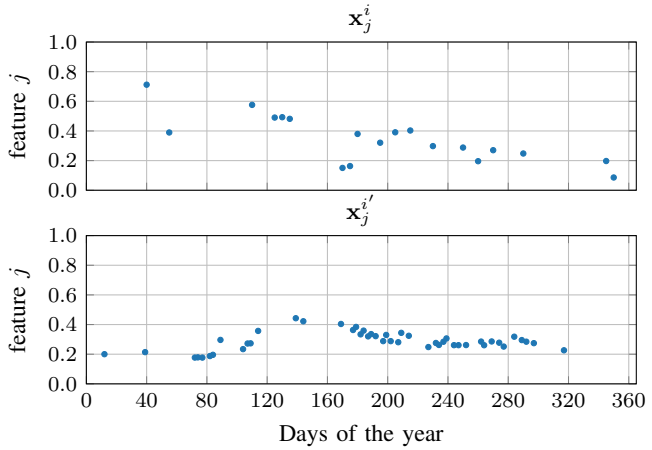



Fig. 2: $\mathbf{x}_j^i$ and $\mathbf{x}_j^{i'}$ are two irregular and unaligned time series for respectively the pixel $i$ and $i'$ for the spectral feature $j$.

$$m^i(t_k) = \begin{cases} 1 & \text{if } t_k \in \mathbf{T}^i \\ 0 & \text{otherwise} \end{cases} \quad \forall t_k \in \mathbf{T}, \tag{1}$$

which indicates if the feature $j$ of pixel $i$ at time $t_k$ is observed or not. We further define an *augmented* pixel time series $\mathbf{x}_j^{i*}$ as the pixel

$$x_j^{i*}(t_k) = \begin{cases} x_j^i(t_k) & \text{if } m^i(t_k) = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall t_k \in \mathbf{T}, \tag{2}$$

Using (1) and (2) will simplify the presentation of the interpolator in the following section.

### B. Projection onto a regular-temporal grid

As described previously, most of the classifiers are not able to deal with irregular and unaligned time series. Thus, the core idea is to learn a projection of these irregular and unaligned time series onto a regular temporal grid of $R$ latent dates: $\mathbf{R} = \{r_1, ..., r_l, ..., r_R\}$. As explained in Section I, a large variety of methods were proposed in the remote sensing literature. In this work, we focused on conventional Nadaraya-Watson kernel smoother [20, Chapter 6], because it leads to an efficient interpolation as discussed in the next section.

For a given pixel time series $\mathbf{x}_j^*$, the interpolated $\hat{x}_j$ at latent timestamp $r_l$ using a kernel smoother is given by[1]:

$$\hat{x}_j(r_l) = \frac{\sum_{t_k=t_1}^{t_T} K(r_l, t_k) m(t_k) x_j^*(t_k)}{\sum_{t'_k=t_1}^{t_T} K(r_l, t'_k) m(t'_k)} \tag{3}$$

with $K$ some similarity kernel [20, Chapter 6]. Usually, the RBF kernel is used $K(r_l, t_k) = \exp\left(d(r_l, t_k)\right)$ with $d(r_l, t_k) = -\sigma^{-2}(r_l - t_k)^2$. From (3), $\hat{x}_j(r_l)$ is a convex combination of original pixel values, whose weights are computed using a similarity kernel applied on the temporal domain. With a RBF kernel, the isotropic distance between pixels is computed in the temporal domain thus the similarity is a decreasing function of the temporal distance. Moreover, the parameter $\sigma$, learned from the training data, weights the temporal distance.

The performances of such method are strongly limited by the hand-crafted similarity kernel. A powerful extension is obtained using *attention* and *embedding* mechanisms, which are able to build more complex similarity kernel [21, Chapter 11]. In the following, the Multi Time Attention Networks (mTAN) [18] is discussed as an extension of the kernel smoother to build the interpolator for the classification model in our end-to-end training.

[1]For clarity, we consider only one pixel and we drop the index $i$ in the remaining of the paper.

### C. Multi Time Attention Networks (mTAN)

To construct the similarity kernel, Shukla et al. [18] proposed to use attention mechanisms and more precisely the scaled-dot product attention. As defined in [21], by using attention mechanisms, "*the neural network can select elements from a set and construct an associated weighted sum over representations*". To compute the scaled dot-product attention, firstly a dot product between a query $q$ and a key $k$ both of size $d_k$ is applied [22]. Then, this dot product is scaled down by $d_k$. Finally, the result is passed through a softmax operation and is multiplied by the value $v$ of size $d_v$ such as:

$$\text{Attention}(q, k, v) = \text{softmax}(\frac{qk^\top}{\sqrt{d_k}}) \times v \tag{4}$$

Different representations can be used for the query $q$ and the key $k$. The mTAN uses a learnable time embedding function (namely *positional encoding*) $\phi$ to map a given $t$ onto a higher dimensional space of size $E$:

$$\begin{aligned} \phi : \mathbb{R} &\to \mathbb{R}^E \\ t &\mapsto \phi(t) = \begin{bmatrix} \omega_1 t + \alpha_1 \\ \sin(\omega_2 t + \alpha_2) \\ \vdots \\ \sin(\omega_E t + \alpha_E) \end{bmatrix} \end{aligned} \tag{5}$$

with $\omega_p$ and $\alpha_p$, $p \in \{1, ..., E\}$, the learnable parameters.

Therefore, to construct the similarity kernel $K$ in (3), we define:

$$d(r_l, t_k) = \frac{\phi(r_l)^\top \mathbf{W}_q^\top \mathbf{W}_k \phi(t_k)}{\sqrt{E}}$$

with $\mathbf{W}_q$ and $\mathbf{W}_k$ two learnable matrices of size $E \times E$, the indices $q$ and $k$ refer to *query* and *key* in (4).
Denoting $\Phi(\mathbf{T}) = \left[\phi(t_1), \ldots, \phi(t_T)\right]$, the matrix of embeddings of $\mathbf{T}$, (3) can be re-written using a masked softmax operator [21, Chapter 11.3.2] such as:

$$\begin{aligned} \hat{x}_j(r_l) &= \text{softmax}\left\{ \frac{\left(\Phi(\mathbf{T})^\top \mathbf{W}_k^\top \mathbf{W}_q \phi(r_l)\right) \odot \mathbf{m}}{\sqrt{E}} \right\}^\top \mathbf{x}_j^* \\ &= \boldsymbol{\gamma}_{r_l}^\top \mathbf{x}_j^*. \end{aligned} \tag{6}$$

with $\odot$ being the Hadamard product, $\mathbf{x}_j^*$ refers to *value* in (4). Authors of [18] further propose to use multi-head attention, i.e., $H$ matrices of embeddings with $\Phi_H(\mathbf{T}) = \{\Phi_h(\mathbf{T})\}_{h=1}^H$, and also $H$ time embedding functions with

$\boldsymbol{\phi}_H(r_l) = [\phi_1(r_l), ..., \phi_H(r_l)]$. A learnable layer $\boldsymbol{\beta}_H$ of size $1 \times H$ is used to produce the interpolated value

$$\hat{x}_j(r_l) = \boldsymbol{\beta}_H(\boldsymbol{\gamma}_{r_l}^H)^\top \mathbf{x}_j^*. \tag{7}$$

This equation can be computed for every spectral feature $j$ and every latent date $r_l$.

The mTAN, as defined in (7), has extended interpolation flexibility w.r.t. the conventional kernel smoother. Also, it is worth noting that (7) benefits from the computational efficiency of attention mechanism (parallel computation) and all parameters are learnable during the training step.

## III. Adaptation of the mTAN for the classification model in our end-to-end learning

In this paper, we propose to use end-to-end learning by combining the mTAN $h_{\boldsymbol{\theta}_1}$ described in Section II-C with a classifier $f_{\boldsymbol{\theta}_2}$ as defined in Fig. 1. The classifier proposed is the Sparse Variational Gaussian Processes (SVGP) defined in [12]. This classifier uses kernel functions, i.e. RBF covariance functions, and no changes were made from [12] (i.e. same loss). This section presents how the mTAN is modified in order to improve the representation obtained for the classification task.

### A. Spectro-temporal feature reduction

The mTAN interpolation allows to perform feature reduction, in the temporal domain, in the spectral domain or in both of them. Indeed, the interpolated feature $j$ is of size $R$ and by taking $R < T$ we can perform a temporal feature reduction. Futhermore, adding a linear layer after the interpolation, spectral feature reduction can done. Noting $\hat{\mathbf{x}}(r) \in \mathbb{R}^D$ the vector of all interpolated spectral features at timestamp $r_l$, $\mathbf{B}$ a matrix of size $D' \times D$ with $D' < D$, the final latent interpolated pixel $\mathbf{z}(r_l)$ can be written as

$$\mathbf{z}(r_l) = \mathbf{B}\hat{\mathbf{x}}(r_l) \tag{8}$$

The overall spectro-temporal feature reduction can be written as:

$$\mathbf{Z} = \mathbf{B}\mathbf{X}^*\boldsymbol{\Gamma} \tag{9}$$

where $\mathbf{Z} = [\mathbf{z}(r_1), \ldots, \mathbf{z}(r_R)] \in \mathbb{R}^{D' \times R}$, $\mathbf{X}^* = [\mathbf{x}^*(t_1), \ldots, \mathbf{x}^*(t_T)] \in \mathbb{R}^{D \times T}$ and $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_{r_1}, \ldots, \boldsymbol{\gamma}_{r_R}] \in \mathbb{R}^{T \times R}$.

As defined in (9), $\boldsymbol{\Gamma}$ does not depend on spectral band and $\mathbf{B}$ does not depend on time i.e. a matrice $\mathbf{B}$ per date is not required. Thus, as Constantin et al. [15], we have defined an independence hypothesis. Therefore, the spectro-temporal structure of the pixels time-series are used to construct the latent variable $\mathbf{Z}$.

Yet, the spatial information is not taken into account. In the following section, we discuss how the spatial coordinates are integrated in the processing by means of spatial positional encoding, which is different from the temporal positional encoding defined in (5).

### B. Spatial positional encoding

The latent variable $\mathbf{Z}$ defined in (9) take into account the spectro-temporal structure of the data. In this paper, we thus proposed to also add the spatial information in $\mathbf{Z}$ by using the *spatial positional encoding*. As in [19], the spatial coordinates $(\psi_1, \psi_2)$ are mapped onto a higher dimensional space of dimension $F$ using $\varphi$:

$$\begin{aligned} \varphi : \mathbb{R}^2 &\to \mathbb{R}^F \\ (\psi_1, \psi_2) &\mapsto \varphi(\psi_1, \psi_2) \\ &= \Big[\sin(\psi_1\nu_1), \cos(\psi_1\nu_1), \ldots, \cos(\psi_2\nu_{F/4})\Big]^\top \end{aligned}$$

with $\nu_q = 10000^{-(2l)/F}$ and $q \in \{1, ..., F/4\}$. $\varphi(\psi_1, \psi_2)$ is then given to a two layers perceptron with ReLu non-linearities to obtain a vector of size $D$ which is finally duplicated for each timestamp to get a spatial positional encoding matrix $\mathbf{P}$ of same shape than $\mathbf{X}^*$ (i.e. $D \times T$). This matrix is added to the raw input data $\mathbf{X}^*$ before the spectro-temporal interpolation:

$$\tilde{\mathbf{X}}^* = \mathbf{X}^* + \mathbf{P}.$$

The parameters of the perceptron are optimized jointly during the learning step.

By using end-to-end learning, the SVGP classifier $f_{\boldsymbol{\theta}_2}$ computed the similarity over the latent spectro-temporal representations of two pixels respectively noted $\mathbf{Z}^i$ and $\mathbf{Z}^{i'}$ such as:

$$\begin{aligned} k(\mathbf{Z}^i, \mathbf{Z}^{i'}) &= \alpha \exp\left(-\frac{\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_F^2}{2\sigma^2}\right) \\ &= \alpha(k^{\lambda t}(\mathbf{x}^i, \mathbf{x}^{i'}) \times k^{\psi}(\{\psi_1^i, \psi_2^i\}, \{\psi_1^{i'}, \psi_2^{i'}\}) \\ &\quad \times k^{\lambda t \psi}(\mathbf{x}^i\{\psi_1^i, \psi_2^i\}, \mathbf{x}^{i'}\{\psi_1^{i'}, \psi_2^{i'}\})) \end{aligned}$$

Details of the calculation can be found in Appendix A. By comparison to our previous works [12], the covariance function $k(\mathbf{Z}^i, \mathbf{Z}^{i'})$ is composed of an additional element: a spatio-spectro-temporal function $k^{\lambda t \psi}$. Indeed, in [12], the covariance function was only the product between a spectro-temporal covariance function $k^{\lambda t}$ and a spatial covariance function $k^{\psi}$. By using *spatial positional encoding*, we have a supplementary source of information.

By using spectro-temporal feature reduction and spatial positional encoding, the mTAN is now designed to produce a suitable latent representation $\mathbf{Z}$ for the classification task. We will now discuss on the different learnable parameters.

### C. Description of the parameters

The parameters $\boldsymbol{\theta}_1$ of the mTAN $h_{\boldsymbol{\theta}_1}$ and their corresponding sizes are summarized in the Table I. Thus, the total number of learnable parameters $\boldsymbol{\theta}_1$ is described by the following equation:

$$2HE(1 + E) + DD' + H + L_1(L_2 + D)$$

As a reminder, the parameters $\boldsymbol{\theta}_2$ of the SVGP classifier $f_{\boldsymbol{\theta}_2}$ were highly dependent on the number of spectro-temporal

TABLE I: Description of the mTAN's parameters $\boldsymbol{\theta}_1$ and their corresponding sizes. The MLP corresponds to the parameters of two layers perceptron used to obtain the spatial positional encoding matrix $\mathbf{P}$ described in the previous section.

| Parameters | Size |
|---|---|
| $\{\omega_p, \alpha_p\}_{p=1}^{E}$ | $2(HE)$ |
| $\mathbf{W}_q, \mathbf{W}_k$ | $2(HE^2)$ |
| $\mathbf{B}$ | $D'D$ |
| $\boldsymbol{\beta}_H$ | $H$ |
| MLP | $L_2(L_1 + D)$ |

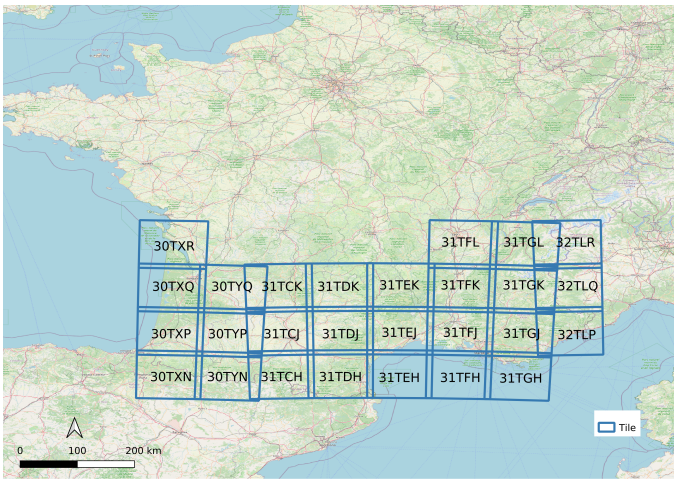



Fig. 3: Location of the 27 studied tiles where a blue square corresponds to one tile as provided by the Theia Data Center[2]. Each tile is displayed with its name in the Sentinel-2 nomenclature.(background map © OpenStreetMap contributors)

features $T \times D$. By using an end-to-end training with the mTAN, this number is significantly reduced to $R \times D'$ with $R < T$ and $D' < D$ and therefore the total number of parameters $\boldsymbol{\theta}_2$ is reduced as well. All the parameters are optimized using a loss function for classification.

## IV. Experimental set-up

The study area covers a zone of approximately $200\,000\,\mathrm{km}^2$ in the south of metropolitan France. It is composed of 27 Sentinel-2 tiles, as displayed in Fig. 3.

### A. Irregular and unaligned data sets

All available acquisitions of level 2A between January and December 2018 for the 27 Sentinel-2 tiles were used, as the ones described in [12] . Indeed, surface reflectance time-series and cloud/shadow masks were produced using the MAJA preprocessing chain [23]. All the bands at 20m/pixel were spatially up-sampled to 10m/pixel using bicubic interpolation [24]. A total of 10 spectral bands with three spectral indices (NDVI, NDWI, Brightness) were used. However, in this paper, no temporal sampling pre-processing was used (i.e. no linear interpolation such as in [7] or other types of temporal synthesis). Therefore, as described in Section I, the data obtained is irregular and unaligned. Following the notations defined in Section II-C, the union of the acquisition dates of the 27 tiles results in $T = 303$ dates. Besides, the spectral dimension is equal to $D = 13$.

[2]https://www.theia-land.fr/en/products/

TABLE II: Number of pixels for each data set

| Training | Validation | Test |
|---|---|---|
| 92 000 | 23 000 | 230 000 |

TABLE III: Land cover classes used for the experiments with their corresponding color code.

| Color | Code | Name |
|---|---|---|
| | CUF | Continuous urban fabric |
| | DUF | Discontinuous urban fabric |
| | ICU | Industrial and commercial units |
| | RSF | Road surfaces |
| | RAP | Rapeseed |
| | STC | Straw cereals |
| | PRO | Protein crops |
| | SOY | Soy |
| | SUN | Sunflower |
| | COR | Corn |
| | RIC | Rice |
| | TUB | Tubers / roots |
| | GRA | Grasslands |
| | ORC | Orchards and fruit growing |
| | VIN | Vineyards |
| | BLF | Broad-leaved forest |
| | COF | Coniferous forest |
| | NGL | Natural grasslands |
| | WOM | Woody moorlands |
| | NMS | Natural mineral surfaces |
| | BDS | Beaches, dunes and sand plains |
| | GPS | Glaciers and perpetual snows |
| | WAT | Water bodies |

The reference data used in this work is composed of $C = 23$ land cover classes ranging from artificial areas to vegetation and water bodies constructed with different data sources as described in [12]. The nomenclature of the 23 land cover classes can be found in Table III.

Pixels were randomly sampled from these polygons over the full study area (i.e. 27 tiles) to create three *spatially disjoint* data subsets: *training*, *validation* and *test*. The three data sets are class-balanced: 4 000 pixels per class in the *training* data set, 1 000 pixels per class in the *validation* data set and 10 000 pixels per class in the *test* data set. The total number of pixels for each data set is provided in Table II. Classification metrics such as overall accuracy (OA) or F-score were computed for each model using the *test* data set with 9 runs with different random pixel samplings. Standardization was performed for the valid acquisitions dates. Mean and standard deviation were estimated for each spectral band and for each spectral index on the *training* data set and then used to standardize the others data sets (*training*, *validation*, *test*) [25].

### B. Competitive methods

As described in [12], the SVGP model had satisfactory classification performance results but feature extraction on the spectro-temporal features is required in order to better take into account the spectro-temporal structure and also to reduce the number of spectro-temporal features. By using end-to-end training, with the mTAN $h_{\boldsymbol{\theta}_1}$ described in Section III and with the SVGP model defined as the classifier $f_{\boldsymbol{\theta}_2}$, spectro-temporal

reduction and spatial positional encoding are considered. In order to evaluate the classification's performances, in addition to the SVGP model, two other classifiers $f_{\boldsymbol{\theta}_2}$ are studied:

- Multi-layer Perceptron (MLP) with the same setup as in [12],
- Lightweight Temporal Self-Attention (LTAE) described in [11].

The end-to-end training models are respectively called *mTAN-SVGP*, *mTAN-MLP* and *mTAN-LTAE*.

Unlike SVGP or MLP classifiers, the LTAE classifier uses attention mecanisms. It may appear redundant to use attention mecanisms both in the mTAN and in the LTAE. Therefore, the LTAE classifier was also studied without the mTAN and this method is called *raw-LTAE*. However, the LTAE classifier is not able to directly deal with the irregular and unaligned time serie pixels. In order to help this classifier, the mask was used as an additionnal feature. Besides, the spatial positional encoding matrix $\mathbf{P}$ was also used in this classifier.

To have a comparison with the *mTAN-SVGP* model, linearly interpolated data was feed into a simple SVGP classifier called *Gapfilled-SVGP* model.

The optimizer parameters for each model were found by trial and error and are described in Table VIII in Appendix B. To process the RF model, 20 CPU with a total of RAM of 100 GB were used and one NVIDIA Tesla V100 GPU, for all other models.

## V. Study of mTAN-SVGP

This section presents the different results obtained from the *mTAN-SVGP* model. Firstly, the performances of the *mTAN-SVGP* are evaluated with different configurations. Then, the latent representation obtained from the mTAN is studied as well as the versatility of its similarity kernel.

### A. Performance results

*1) Comparison with linear interpolation:* Firstly, the *mTAN-SVGP* was implemented with a vector of latent dates $\mathbf{R}$ defined with a regular sampling of $\tau = 10$ days and a total number of $R = 37$ dates[3]. Moreover, the number of latent spectral features was equal to the number of spectral features such as $D' = D = 13$. The latent representation $\mathbf{Z}$ obtained using the mTAN is described by $R \times D' = 481$ spectro-temporal features. The *Gapfilled-SVGP* model was implemented with the same number of spectro-temporal features. A detailed evaluation of performance of the *Gapfilled-SVGP* model was done in [12] including a comparison with the Random Forest (RF).

As shown in Table IV, the *mTAN-SVGP* model is 10 points above the *Gapfilled-SVGP* model. The latent representation $\mathbf{Z}$ obtained with the mTAN extracts more meaningful information for the SVGP classifier compared to the linearly interpolated data.

[3]Experiments were also made with random irregular sampling and with selected samples from cumulative histograms. As modifying the positions of the latent dates do not have any influence on the performances, the simpliest method was selected: regular sampling.

TABLE IV: Averaged overall accuracy (OA) for the *mTAN-SVGP* and *Gapfilled-SVGP* models (mean % ± standard deviation computed with 9 runs)

| mTAN-SVGP | Gapfilled-SVGP |
|---|---|
| 77.44 ± 0.15 | 67.25 ± 0.37 |

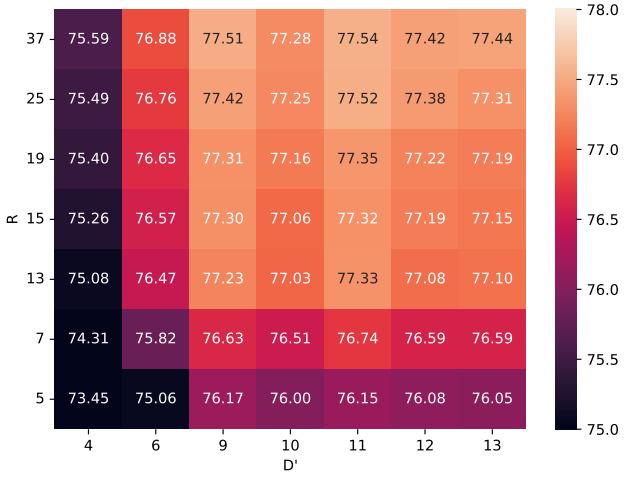



Fig. 4: Averaged overall accuracy (OA) for $H = 1$ (mean in % computed over 9 different runs) with $R$ the number of latent dates and $D'$ the number of latent spectral features.

*2) Spectral and temporal reduction:* Fig. 4 and Fig. 5 represent respectively the averaged overall accuracy (OA) and the averaged training times computed with different number of latent dates $R = \{5, 7, 13, 15, 19, 25, 37\}$ and different number of latent spectral features $D' = \{4, 6, 9, 10, 11, 12, 13\}$ in the mTAN. As shown in Fig. 4, reducing $R$ from 37 to 13 and $D'$ from 13 to 9 has a negligible effect on the OA (i.e. from 77.44 to 77.23). Moreover, the variance of the OA is small, around 0.15. The number of parameters $\boldsymbol{\theta}_2$ is almost reduced by a factor 4 (i.e. from 584 200 to 165 600 parameters) and the training times are divided by two, as described in Fig. 5.

In addition, the number of heads $H$ has little impact on the classification performances as shown in Fig. 13 in Appendix C. Besides, from $H = 1$ to $H = 3$, the training time can be increased by a factor of 2 as shown in Fig. 14 in Appendix C.

*3) Spatial positional encoding:* The spatial information used to compute the positional encoded matrix $\mathbf{P}$ is composed of the spatial coordinates (northing $\psi_1$ and easting $\psi_2$) in meters in the Lambert 93 projection. The number of neurons in the first and second layer are respectively $L_1$ and $L_2$ and were found by trial and error: $L_1 = 16$ and $L_2 = 14$.

As shown in Table V, the use of the spatial positional encoding in the mTAN for the *mTAN-SVGP* model increased by nearly 1.5 points the overall accuracy.

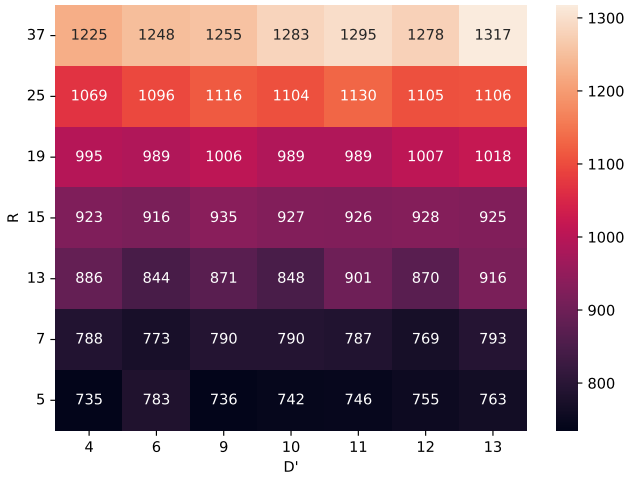



Fig. 5: Averaged training times for $H = 1$ (mean in sec computed over 9 different runs) with $R$ the number of latent dates and $D'$ the number of latent spectral features.

TABLE V: Averaged overall accuracy (OA) without (standard) and with the spatial positional encoded matrix (with $\mathbf{P}$) in the mTAN for the model *mTAN-SVGP* (mean $\%$ $\pm$ standard deviation computed with 9 runs)

| **Standard** | **With P** |
|---|---|
| 77.23 $\pm$ 0.17 | 78.63 $\pm$ 0.16 |

Fig. 6 represents the value $\mathbf{P}$ respectively for the features 4 and 12. This value was computed using 3234 different spatial coordinates on a regularly spaced grid over the 27 tiles. As shown in both Fig. 6a and 6b, the spatial transitions are quite smooth. Each feature takes into account differently the spatial information.

*4) Influence on the number of inducing points:* Fig. 7 represents the number of learnable parameters $\boldsymbol{\theta}_2$ based on the number of spectro-temporal features $R \times D'$ and the number of inducing points $M$. By using spectro-temporal reduction as described in Section V-A2, the number of spectro-temporal features has been considerably reduced from 481 ($R = 37, D' = 13$) to 117 ($R = 13, D' = 9$). It results in a significant reduction for $\boldsymbol{\theta}_2$ as shown in Fig. 7. Moreover, by doubling the number of inducing points from 50 to 100, the number of parameters $\boldsymbol{\theta}_2$ with $R = 13, D' = 9$ is still lower than with 50 inducing points and $R = 37, D' = 13$. Experiments were done by increasing the number of inducing points $M = \{100, 150, 200\}$. Table VI represents the averaged overall accuracy and training times computed with different number of inducing points. With $M = 200$, the overall accuracy is almost increased by one point compared to $M = 50$. Training time is only slightly affected by this increase in the number of inducing points, i.e. 834 to 967. Thus, spectro-temporal reduction had made possible the reduction of the number of parameters. Above all, it reduced the complexity, which is highly proportional to the correlations between the variables.

TABLE VI: Averaged overall accuracy (OA) (mean $\%\pm$ standard deviation) and averaged training times (in sec) for the *mTAN-SVGP* with $R = 13$ latent dates, $D' = 9$ latent spectral features, $H = 1$ head and the spatial positional encoded matrix $\mathbf{P}$ for different number of inducing points $M$ (computed over 9 runs).

| | **Number of inducing points M** | | | |
|---|---|---|---|---|
| | **50** | **100** | **150** | **200** |
| Averaged OA | 78.63 $\pm$ 0.16 | 79.20 $\pm$ 0.21 | 79.43 $\pm$ 0.29 | 79.48 $\pm$ 0.17 |
| Training time | 834 | 910 | 921 | 967 |

### *B. Latent representation obtained from the mTAN*

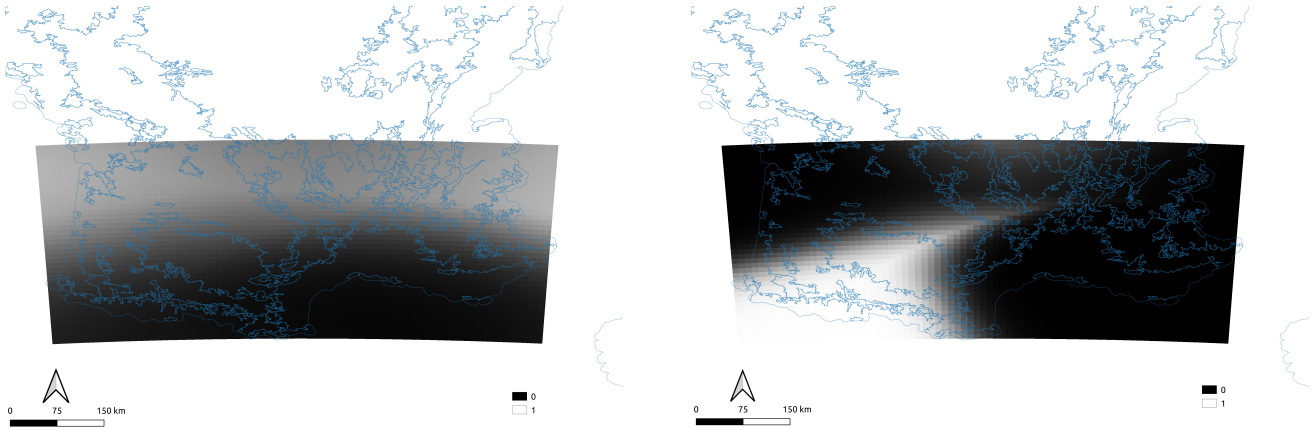
The latent representation $\hat{\mathbf{x}}_j$ obtained from the mTAN from (7) with the *mTAN-SVGP* model minimises the classification error. In order to get an idea of this representation obtained, illustrations are provided in Fig. 8. The latent representation $\hat{\mathbf{x}}_j$ is projected onto a regular grid thus it can be easily visualized with the raw data and with the gapfilled data i.e. linearly interpolated data. Fig. 8 represents the comparison of these three time series profiles (raw, gapfilled and mTAN) for one pixel labeled as "CORN". The spectral feature $j$ is not modified (i.e. linear mixing has not yet been applied) and corresponds to the NDVI.

The latent mTAN representation obtained in Fig. 8 is similar to the raw data or to the gapfilled data even if it do not minimize the reconstruction error. It retains only essential information hence the small number of latent dates $R$ in the previous section.

### *C. Versatility of the similarity kernel*

As defined in Section II-C, by using *attention* and *embedding* mechanisms, the similarity kernel can be written such as: $K(r_l) = \exp\left(d(r_l)\right)$ with $d(r_l) = \frac{\left(\Phi(\mathbf{T})^\top \mathbf{W}_k^\top \mathbf{W}_q \phi(r_l)\right) \odot \mathbf{m}}{\sqrt{E}}$. This kernel is able to adapt to the pixel sampling for the classification task. The versatility of the similarity kernel can be shown by computing the attention value $\boldsymbol{\gamma}_{r_l}$ defined in (6) for different latent dates $r_l$ and for different sets of observed dates $\mathbf{T}$. Fig. 9 represents the normalized attention values $\boldsymbol{\gamma}_{r_l}^n$ computed for three different latent dates: $r_l \in \{1, 181, 361\}$ and with three different sets of observed dates $\mathbf{T}$. The first one, in red in both figures, is computed with $\mathbf{T} = \{1, .., 365\}$ with a regular interval of $\tau = 1$ day. The last two, respectively in blue and green in Fig. 9a and 9b, are computed with two different temporal grids ($\mathbf{T}^i \neq \mathbf{T}^{i'}$) corresponding to two random pixels $i$ and $i'$.

As shown in Fig. 9a and 9b, the kernel is not centered on the query date $r_l$. It adapts itself according to the latent date $r_l$. Moreover, as shown in Fig. 9, for the set of observed dates $\mathbf{T} = \{1, .., 365\}$ (i.e. in red), the bandwidth is larger for the latent date $r_l = 1$ than for the latent date $r_l = 181$. The kernel can be described as heteroscedastic: for different latent dates $r_l$, the bandwidth of our kernel varies.




(a) feature 4 (b) feature 12

Fig. 6: Spatial positial encoding $\mathbf{P}$ computed using 3234 different spatial coordinates on a regularly spaced grid over the 27 tiles

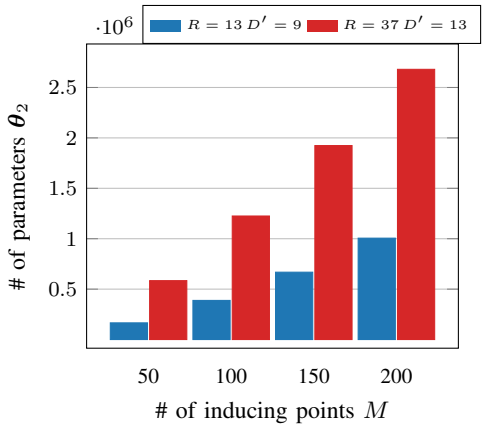



Fig. 7: Number of learnable parameters $\boldsymbol{\theta}_2$ based on the number of inducing points $M$ and the number of spectro-temporal features $R \times D'$.

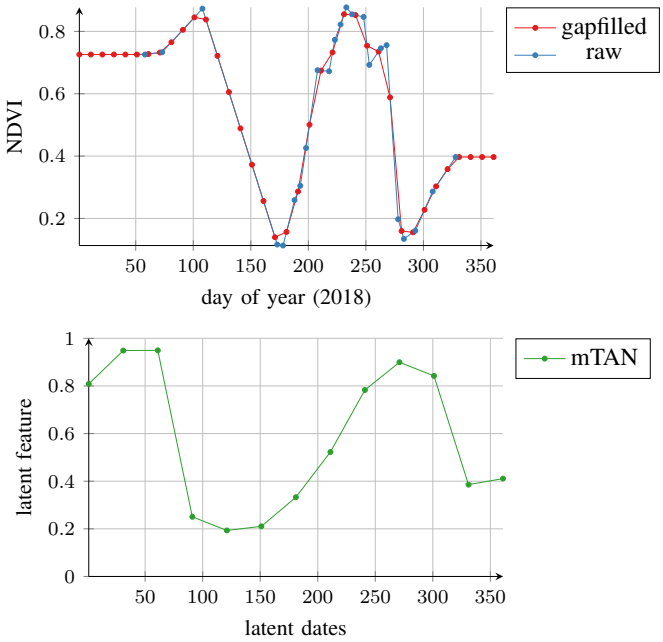



Fig. 8: NDVI time series profiles for a pixel labeled "CORN". — raw corresponds to the raw data, the outlier values have been removed in order to have a comprehensive plot. — gapfilled corresponds to the value obtained with a linear interpolation with an interval of 10 days for a total of 37 dates. — mTAN corresponds to the mTAN representation $\hat{\mathbf{x}}_j$ with $j$ = NDVI obtained from the *mTAN-SVGP* model, before the spectral reduction ($D' = 9$).

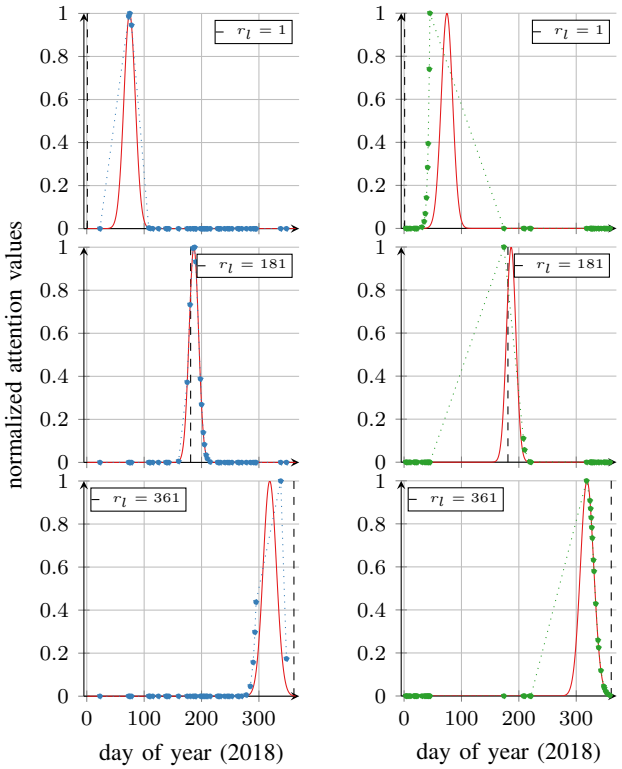
## VI. Comparison with competitive methods

This section presents a comparison of the *mTAN-SVGP* model described in the previous section with different models: *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*. Firstly, the performance results are studied quantitatively and qualitatively. Then, a further comparative study is made between the *mTAN-SVGP* and the *raw-LTAE* concerning the temporal sampling.

### A. Performance results

The *mTAN-SVGP* model is compared with 3 different models: *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*. In the following sections, from the results obtained in the previous section V and with the best compromise between time and performance, the mTAN is defined with $R = 13$ latent dates, $D' = 9$ latent spectral features, $H = 1$ head, the use of the spatial positional encoding matrix $\mathbf{P}$ and $M = 200$ inducing points. The *raw-LTAE* model is the only one not using the mTAN. Thus, no spectral or temporal reduction was implemented in this model.

*1) Quantitative results:* As shown in Fig. 10, the SVGP model took more advantage of the mTAN than the MLP or the LTAE models. Indeed, the overall accuracy of the *mTAN-SVGP* model is seven points above the *mTAN-MLP* model and around four points above the *mTAN-LTAE* model. On the other hand, *mTAN-SVGP* model is in average two points below the




(a) $i$th random pixel (b) $i'$th random pixel

Fig. 9: Normalized attention values $\gamma_r^n$ for three different latent dates: $r_l \in \{1, 181, 361\}$. — corresponds to $\gamma_r^n$ computed with $\mathbf{T} = \{1, .., 365\}$ with a regular interval of $\tau = 1$ day. • and • correspond to $\gamma_r^n$ respectively computed with two different temporal grids ($\mathbf{T}^i \neq \mathbf{T}^{i'}$) corresponding to two random pixels $i$ and $i'$.

TABLE VII: Averaged training times (in sec) computed over 9 runs and number of trainable parameters for each model (*mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*).

| | **mTAN-SVGP** | **mTAN-MLP** | **mTAN-LTAE** | **raw-LTAE** |
|---|---|---|---|---|
| Training time | 967 | 1207 | 840 | 1279 |
| # parameters | 1 005 675 | 33 113 | 184 376 | 761 380 |

*raw-LTAE* model.

The number of trainable parameters and the training times for each method are summarized in Table VII. The *mTAN-SVGP* model has more trainable parameters that the *raw-LTAE* model. However, the training time of the *mTAN-SVGP* model is about 1.3 times shorter than the *raw-LTAE* as shown in Table VII. By using a spectro-temporal reduction with the mTAN, the number of trainable parameters for the *mTAN-SVGP* is just over 2.5 times lower than the simple SVGP (i.e. 1 005 675 versus 2 680 075), as described in Fig. 7. The number of parameters of the *raw-LTAE* is also very large because it is not able to deal with unaligned time series and therefore obliged to combine all the dates. By using the mTAN, for the LTAE, the number of trainable parameters is reduced by four. However, as shown in Fig. 10, the overall accuracy of the *mTAN-LTAE* model is seven points below the *raw-LTAE* model.

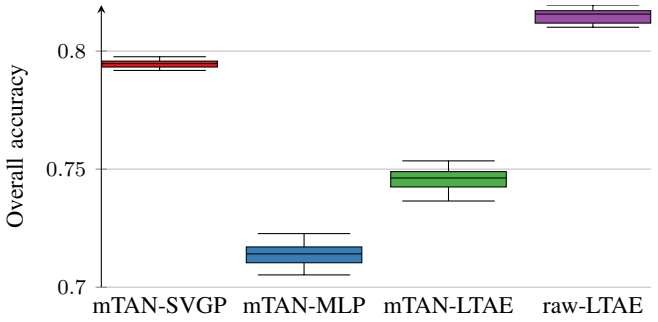



Fig. 10: Boxplots of the overall accuracy for each model (*mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*) computed over 9 runs.

*2) Qualitative results:* Land cover maps have been produced for each model (*mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE*, *raw-LTAE*) using the `iota`[2] processing chain [26] on two different tiles: 31TCJ and 31TDJ. Inference was performed using the model trained on the 27 tiles with the best overall accuracy over the 9 runs. The results obtained for respectively the *mTAN-SVGP* and *raw-LTAE* are shown in Fig. 11a and 11b. The results obtained on this agricultural area on the 31TCJ tile show that the pixels are more homogeneous (with less salt and pepper classification noise [27]) with the *mTAN-SVGP* compared to the *raw-LTAE*. All the land cover maps generated are available for download. [4]

### B. Versatility to the temporal sampling

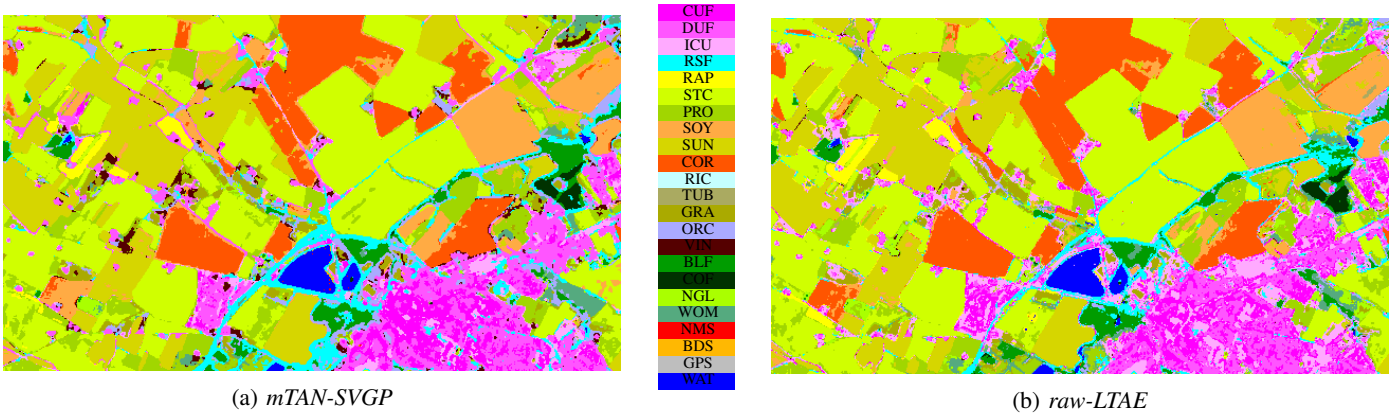
The *raw-LTAE* showed better classification performances. However, to compute the inference on a specific area (e.g. on a specific tile), the *raw-LTAE* required the whole set of observed dates $\mathbf{T} = \{t_1, ..., t_T\}$ available for training. It is not the case of the classifier using the mTAN. Thus, once trained, it is able to classify any irregular and unaligned pixel time series.

An other interesting feature is its capacity for generalization. Indeed, we computed the overall accuracy only on 31TCJ tile for two models *mTAN-SVGP* and *raw-LTAE* both trained on the 27 tiles. The acquisition dates $\mathbf{T}$ for the *test* data set were artificially shifted with different values: $\delta = \{0, 1, 2, 3, 5\}$ days. This shift was chosen because the acquisition dates between pixels on two close orbits are shifted by a maximum of five days. As shown in Fig. 12, the overall accuracy of the *mTAN-SVGP* model is not affected by this temporal shift $\delta$. However, the overall accuracy of the *raw-LTAE* model is greatly impacted by the temporal shift $\delta$ and is almost divided by 1.5 with $\delta = 5$ days. By using a linear smoother with temporal attention mecanisms, the *mTAN-SVGP* model is more robust to this shift than the *raw-LTAE* model which use spectro-temporal attention mecanisms.

## VII. Conclusions and Perspectives

This work introduces an approach using irregular and unaligned Sentinel-2 SITS for large-scale land cover pixel-based

[4]DOI: https://doi.org/10.5281/zenodo.8033902




(a) *mTAN-SVGP* (b) *raw-LTAE*

Fig. 11: Land cover maps obtained on an agricultural area on the tile 31TCJ

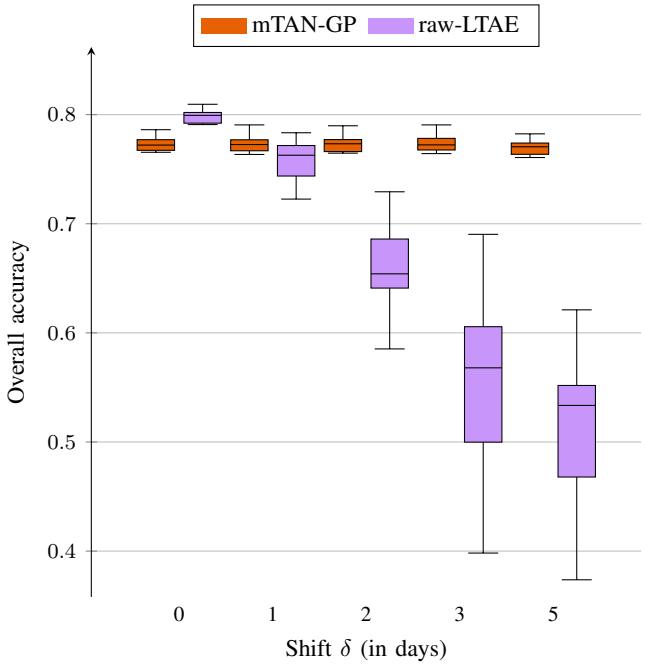



Fig. 12: Boxplots of the overall accuracy for the *mTAN-SVGP* and *raw-LTAE* models computed with the *test* data set only on the 31TCJ tile over 9 runs. The models were trained and validated on the all 27 tiles. The acquisition dates $\mathbf{T}$ for the test data set were artificially shifted with different values: $\delta = \{0, 1, 2, 3, 5\}$ days.

classification. To deal with irregular and unaligned pixel times series end-to-end learning is used. A first module, the Multi-Attention Time Networks (mTAN), enables to project the irregular and unaligned SITS onto a fixed and reduced size representation. This representation is then given to the SVGP classifier and all the parameters are optimized using a loss function for classification. The spatial information is taken into account in the representation through the *spatial positional encoding*. Experiments were conducted on Sentinel-2 SITS of the full year 2018 in an area of 200 000 km$^2$ in the south of France. In terms of accuracy, the end-to-end learning *mTAN-SVGP* model outperformed the simple SVGP classifier with linearly interpolated data (*Gapfilled-SVGP*). The significant reduction for the spectro-temporal features has allowed to use more inducing points while keeping the same complexity, resulting in improved classification performance. Moreover, the *mTAN-SVGP* model is above the *mTAN-MLP* and *mTAN-LTAE* models in terms of accuracy.

In this paper, the potential of the multi-head attention has not been fully taken into account. Indeed, only one head was used $H = 1$ and the performances with an increasing number of heads were not satisfying. A perspective of this work is for each head to specialize using the spatial information. The *spatial positional encoding* could be set up to help the heads to specialize and differentiate themselves.

Recently, the litterature [28]–[30] has shown an improvement in classification performance by combining radar and optical time series (i.e. Sentinel 1 and 2). Thus, a perspective of this work is to add Sentinel-1 time series to the actual time series. In addition, other types of data such as weather data can be added in the mTAN module in order to create a representation for the SVGP classifier. Moreover, in addition to spatial data (i.e. longitude and latitude), topographic data can be used to construct the *spatial positional encoding* in order to take better account of climatic, geographical and other differences.

In the interest of reproducible research, the implementation of all the models (*mTAN-GP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*) is made available in the following repository:https://gitlab.cesbio.omp.eu/belletv/land_cover_southfrance_mtan_gp_irregular_sits.

## ACKNOWLEDGMENT

The authors would like to thank Benjamin Tardy for his support and help during the generation of the different data sets and the production of land cover classification maps with the `iota`$^2$ software. Finally, the authors would also like to thank CNES for the provision of its high performance computing (HPC) infrastructure to run the experiments presented in this paper and the associated help.

## REFERENCES


[1] IPCC, "SYNTHESIS REPORT OF THE IPCC SIXTH ASSESSMENT REPORT (AR6) longer report." https://report.ipcc.ch/ar6syr/pdf/IPCC_AR6_SYR_LongerReport.pdf, 2023.

[2] C. Persello, J. D. Wegner, R. Hansch, D. Tuia, P. Ghamisi, M. Koeva, and G. Camps-Valls, "Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current Approaches, Open Challenges, and Future Opportunities," *IEEE Geoscience and Remote Sensing Magazine*, pp. 2–30, 2022.

[3] D. Tuia, K. Schindler, B. Demir, G. Camps-Valls, X. X. Zhu, M. Kochupillai, S. Džeroski, J. N. van Rijn, H. H. Hoos, F. D. Frate, M. Datcu, J.-A. Quiané-Ruiz, V. Markl, B. L. Saux, and R. Schneider, "Artificial intelligence to advance earth observation: a perspective," 2023.

[4] F. Bertini, O. Brand, S. Carlier, U. Del Bello, M. Drusch, R. Duca, V. Fernandez, C. Ferrario, M. H. Ferreira, C. Isola, V. Kirschner, P. Laberinti, M. Lambert, G. Mandorlo, P. Marcos, P. Martimort, S. Moon, P. Oldeman, M. Palomba, and J. Pineiro, "Sentinel-2 esa's optical high-resolution mission for gmes operational services," *ESA bulletin. Bulletin ASE. European Space Agency*, vol. SP-1322, 03 2012.

[5] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 45–54, 2014.

[6] C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu, "Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas," *Remote Sensing of Environment*, vol. 187, pp. 156–168, 2016.

[7] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes, "Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series," *Remote Sensing*, vol. 9, no. 1, 2017.

[8] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel, "Land cover classification via multitemporal spatial data by deep recurrent neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1685–1689, 2017.

[9] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, 2018.

[10] C. Pelletier, G. Webb, and F. Petitjean, "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series," *Remote Sensing*, vol. 11, p. 523, Mar. 2019.

[11] V. Sainte Fare Garnot and L. Landrieu, "Lightweight temporal self-Attention for classifying satellite images time series," in *Workshop on Advanced Analytics and Learning on Temporal Data*, AALTD, Sept. 2020.

[12] V. Bellet, M. Fauvel, and J. Inglada, "Land cover classification with gaussian processes using spatio-spectro-temporal features," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–1, 2023.

[13] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," 2016.

[14] S. C.-X. Li and B. Marlin, "A scalable end-to-end gaussian process adapter for irregularly sampled time series classification," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, p. 1812–1820, Curran Associates Inc., 2016.

[15] A. Constantin, M. Fauvel, and S. Girard, "Joint Supervised Classification and Reconstruction of Irregularly Sampled Satellite Image Times Series," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, p. 4403913, May 2021.

[16] Z. Lu, T. Leen, Y. Huang, and D. Erdogmus, "A reproducing kernel hilbert space framework for pairwise time series distances," *Proceedings of the 25th International Conference on Machine Learning*, pp. 624–631, 01 2008.

[17] S. N. Shukla and B. M. Marlin, "Interpolation-prediction networks for irregularly sampled time series," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[18] S. N. Shukla and B. Marlin, "Multi-time attention networks for irregularly sampled time series," in *International Conference on Learning Representations*, 2021.

[19] L. Baudoux, J. Inglada, and C. Mallet, "Toward a Yearly Country-Scale CORINE Land-Cover Map without Using Images: A Map Translation Approach," *Remote Sensing*, vol. 13, p. 1060, Mar. 2021.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.

[21] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[23] 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 Sensing*, vol. 11, p. 433, Feb. 2019.

[24] O. D. Team, "Orfeo ToolBox 7.1," Mar. 2020. https://zenodo.org/record/3715021.

[25] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC, 1 ed., July 2019.

[26] J. Inglada, A. Vincent, M. Arias, and B. Tardy, *iota2-a25386*, July 2016. https://doi.org/10.5281/zenodo.58150.

[27] H. Hirayama, R. C. Sharma, M. Tomita, and K. Hara, "Evaluating multiple classifier system for the reduction of salt-and-pepper noise in the classification of very-high-resolution satellite images," *International Journal of Remote Sensing*, vol. 40, no. 7, pp. 2542–2557, 2019.

[28] D. Ienco, R. Gaetano, R. Interdonato, K. Ose, and D. Ho Tong Minh, "Combining sentinel-1 and sentinel-2 time series via rnn for object-based land cover classification," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4881–4884, 2019.

[29] N. Kussul, M. Lavreniuk, and L. Shumilo, "Deep recurrent neural network for crop classification task based on sentinel-1 and sentinel-2 imagery," in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 6914–6917, 2020.

[30] M. Gonzalez-Audicana, S. Lopez-Saenz, M. Arias, I. Sola, and J. Alvarez-Mozos, "Sentinel-1 and sentinel-2 based crop classification over agricultural regions of navarre (spain)," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 5977–5980, 2021.

## APPENDIX A
## SPATIAL POSITIONAL ENCODING

The RBF covariance function over the latent spectro-temporal representations of two pixels respectively noted $\mathbf{Z}^i$ and $\mathbf{Z}^{i'}$ can be written such as:

$$\begin{aligned} k(\mathbf{Z}^i, \mathbf{Z}^{i'}) &= \alpha \exp\left(-\frac{\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_F^2}{2\sigma^2}\right) \\ &= \alpha\left(k^{\lambda t}(\mathbf{x}^i, \mathbf{x}^{i'}) \times k^{\psi}(\{\psi_1^i, \psi_2^i\}, \{\psi_1^{i'}, \psi_2^{i'}\}) \times k^{\lambda t \psi}(\mathbf{x}^i\{\psi_1^i, \psi_2^i\}, \mathbf{x}^{i'}\{\psi_1^{i'}, \psi_2^{i'}\})\right) \end{aligned}$$

with

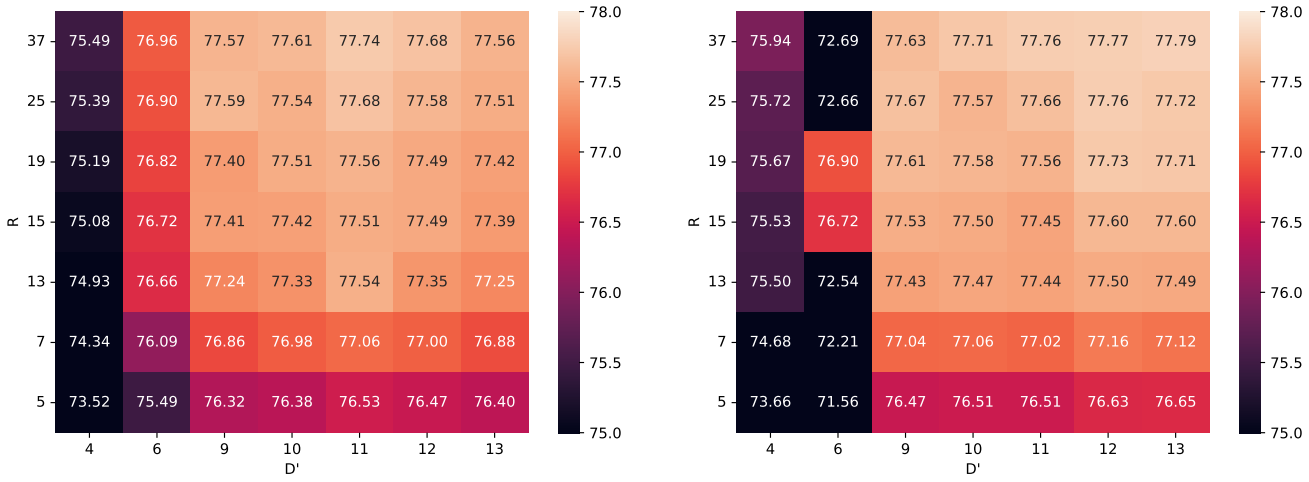
$$\begin{aligned} \|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_F^2 &= \|\mathbf{Z}^i\|_F^2 + \|\mathbf{Z}^{i'}\|_F^2 - 2\langle \mathbf{Z}^i, \mathbf{Z}^{i'}\rangle_F \\ &= \|\mathbf{B}(\mathbf{X}^{i*} + \mathbf{P}^i)\mathbf{\Gamma}^i\|_F^2 + \|\mathbf{B}(\mathbf{X}^{i'*} + \mathbf{P}^{i'})\mathbf{\Gamma}^{i'}\|_F^2 - 2\langle \mathbf{Z}^i, \mathbf{Z}^{i'}\rangle_F \\ &= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i\|_F^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\|_F^2 + 2\langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\rangle_F \\ &\quad + \|\mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_F^2 + \|\mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_F^2 + 2\langle \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_F \\ &\quad - 2\left(\langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_F + \langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_F + \langle \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_F + \langle \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_F\right) \\ &= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_F^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_F^2 \\ &\quad + 2\left(\langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\rangle_F + \langle \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_F - \langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_F - \langle \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_F\right) \\ &= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_F^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_F^2 \\ &\quad + 2\left(\langle \mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}(\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{P}^{i'}\mathbf{\Gamma}^{i'})\rangle_F - \langle \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}, \mathbf{B}(\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{P}^{i'}\mathbf{\Gamma}^{i'})\rangle_F\right) \\ &= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_F^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_F^2 + 2\langle \mathbf{B}(\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}), \mathbf{B}(\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{P}^{i'}\mathbf{\Gamma}^{i'})\rangle_F \end{aligned}$$

## APPENDIX B
## SOLVER PARAMETERS FOR EACH MODEL

TABLE VIII: Parameter values for the Adam optimizer for the models: *Gapfilled-SVGP*, *mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*.

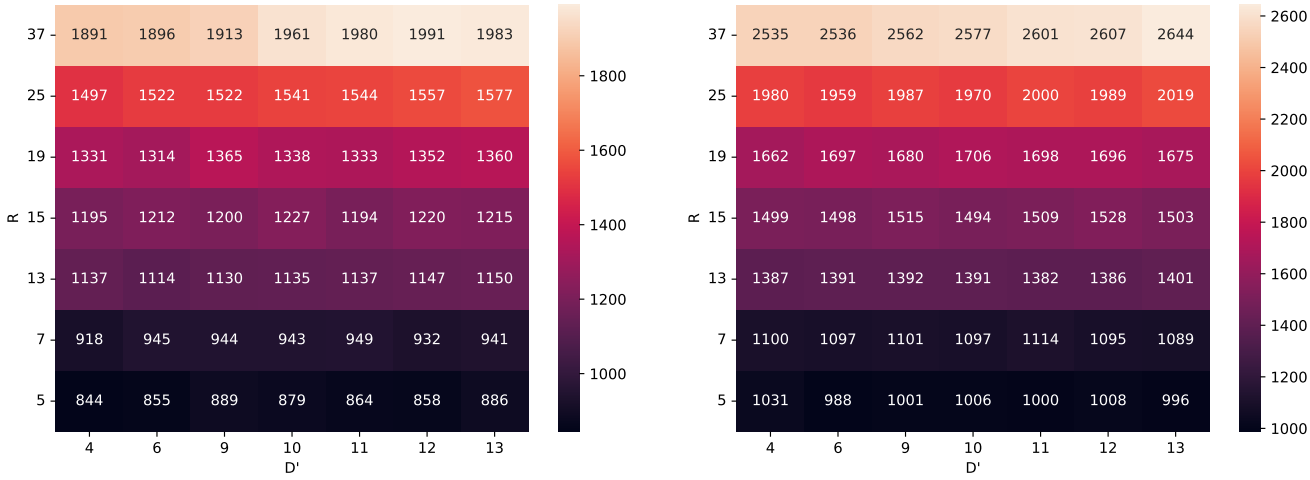
| | **Gapfilled-SVGP** | **mTAN-SVGP** | **mTAN-MLP** | **mTAN-LTAE** | **raw-LTAE** |
|---|---|---|---|---|---|
| Number of epochs | 100 | 100 | 300 | 100 | 100 |
| Batch size | 1024 | 1024 | 1000 | 1000 | 1000 |
| Learning rate | $1 \times 10^{-3}$ | $1 \times 10^{-3}$ | $1 \times 10^{-4}$ | $5 \times 10^{-5}$ | $1 \times 10^{-4}$ |

## APPENDIX C
## INFLUENCE OF THE SPECTRAL AND TEMPORAL REDUCTION FOR DIFFERENT H HEADS




$H = 2$ $H = 3$

Fig. 13: Averaged overall accuracy (OA) (mean in % computed over 9 different runs) with $R$ the number of latent dates, $D'$ the number of latent spectral features and $H$ the number of heads.




$H = 2$ $H = 3$

Fig. 14: Averaged training times (mean in sec computed over 9 different runs) with $R$ the number of latent dates, $D'$ the number of latent spectral features and $H$ the number of heads.