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*—**

***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

II. The irregular and unaligned pixel time series

This section describes how irregular and unaligned pixel time series can be 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, some conventional methods used to project these irregular and unaligned pixel time series are described. Finally, a specific method used in the following of the paper is presented: the mTAN which combines attention mecanisms with multiple continuous time embeddings.

A. Notations and definitions

In this paper, the ith 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 ψ_1^i and ψ_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 the introduction, 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, the Fig. 1 presents two real irregular and unaligned pixel time series acquired with Sentinel-2.

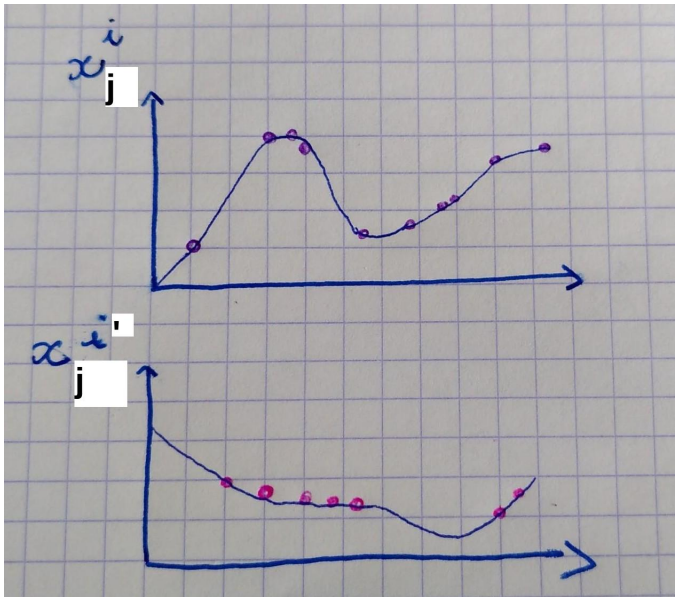


Fig. 1: $\mathbf{x}_j^i$ and $\mathbf{x}_j^{i'}$ are two irregular and unaligned time series for respectively the pixel i and i' for the spectral feature j.

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

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

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

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

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 interpolation layer in the following section.

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)

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 [1, 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 [1, Chapter 6]. Usually, the RBF kernel is used $K(r_l, t_k) = \exp\{-d(r_l, t_k)\}$ 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 an 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 σ, 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 [2, Chapter 11]. In the following, the Multi Time Attention Networks (mTAN) [3] is discussed as an extension of the kernel smoother to build the interpolation layer for the classification model in our end-to-end training.

C. Multi Time Attention Networks (mTAN)

To construct the similarity kernel, [3] proposed to use attention mechanisms and more precisely the scaled-dot product attention. As defined in [2], 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 [4]. 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*) ϕ 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 ω_p and α_p, $p \in \{1, ..., E\}$, the learnable parameters.

Therefore, the similarity kernel can be written such as:

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

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 $\Phi(\mathbf{T}) = \big[\phi(t_1), \ldots, \phi(t_T)\big]$, the matrix of embeddings of $\mathbf{T}$, (3) can be re-written using a masked softmax operator [2, Chapter 11.3.2] such as:

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

with $\odot$ being the Hadamard product, $\mathbf{x}_j^*$ refers to *value* in (4). Authors of [3] 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. **??**. The classifier proposed is the Sparse Variational Gaussian Processes (SVGP) defined in [5]. This classifier uses kernel functions, i.e. RBF covariance functions, and no changes were made from [5] (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 taking $R < T$ we can perform a temporal feature reduction. Futhermore, adding a linear layer after the interpolation,

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

spectral feature reduction can done. Noting $\hat{\mathbf{x}}(r) \in \mathbb{R}^D$ the vector of all interpolated spectral features at timestamp r, $\mathbf{B}$ a matrix of size $D' \times D$ with $D' < D$, the final latent interpolated pixel $\mathbf{z}(r)$ can be written as

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

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

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

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 $\mathbf{\Gamma} = [\boldsymbol{\gamma}_{r_1}, \ldots, \boldsymbol{\gamma}_{r_R}] \in \mathbb{R}^{T \times R}$.

From (9), it is clear how 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 of Eq. (5).

B. Spatial positional encoding

In previous works we have shown that using the spatial coordinates, either for spatial stratification [6] or for learning a spatial-informed classifier [5] leads to better classification accuracy. Indeed, the spatial coordinates help to take into account the non-stationarity in the spectro-temporal domain of the time series.

In this paper, we thus proposed to extend (9) with *spatial positional encoding*. As in [7], the spatial coordinates are mapped onto a higher dimensional space of dimension F using φ:

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

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 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.

As a reminder, in the SVGP classifier, the similarity between two pixels $\mathbf{x}^i$ and $\mathbf{x}^{i'}$ is computed using a RBF covariance function $k(\mathbf{x}^i, \mathbf{x}^{i'})$. By using end-to-end learning, the SVGP classifier $f_{\boldsymbol{\theta}_2}$ computed the similarity over the latent spectro-temporal representations of these two pixels respectively noted $\mathbf{Z}^i$ and $\mathbf{Z}^{i'}$ such as:

$$\begin{aligned} k(\mathbf{Z}^i, \mathbf{Z}^{i'}) &= \alpha \exp\left(-\frac{\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_2^2}{2\sigma^2}\right) \\ &= \alpha(k^{\lambda t}(\mathbf{x}^i, \mathbf{x}^{i'}) \times k^{\psi}(\{\psi_1^i, \psi_2^i\}, \{\psi_1^{i'}, \psi_2^{i'}\}) \\ &\quad \times k^{\lambda t \psi}(\mathbf{x}^i\{\psi_1^i, \psi_2^i\}, \mathbf{x}^{i'}\{\psi_1^{i'}, \psi_2^{i'}\})) \end{aligned}$$

Therefore, the obtained covariance function is similar to the product of a spectro-temporal covariance function $k^{\lambda t}$, a spatial covariance function k^{ψ} and a spatio-spectro-temporal function $k^{\lambda t \psi}$. Details of the calculation can be found in Appendix A. By using the spatial positional encoding, the covariance function $k(\mathbf{Z}^i, \mathbf{Z}^{i'})$ is composed of an additional element compared to the covariance function defined in [5] which was the product between a spectro-temporal covariance function $k^{\lambda t}$ and a spatial covariance function $k^{\lambda t \psi}$. The spatio-spectro-temporal function $k^{\lambda t \psi}$ is thus a supplementary source of information.

By using spectro-temporal feature reduction and spatial positional encoding, the mTAN is now designed to produce a suitable representation 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 the parameters $\boldsymbol{\theta}_2$ of SVGP classifier $f_{\boldsymbol{\theta}_2}$ are optimized using a loss function for classification.
The parameters $\boldsymbol{\theta}_1$ of the mTAN and their corresponding sizes are summarized in the Table I:

TABLE I: Description of the mTAN's parameters $\boldsymbol{\theta}_1$ and their corresponding sizes.

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

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.

As a reminder, the parameters $\boldsymbol{\theta}_2$ of the SVGP, defined in **??**, were highly dependent on the number of spectro-temporal features $d = T \times D$. 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 is reduced as well.

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. 2.

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

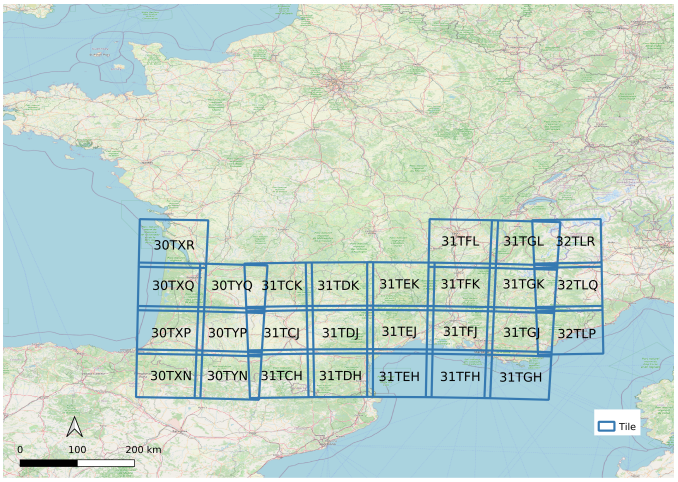


Fig. 2: Location of the 27 studied tiles where a blue square corresponds to one tile as provided by the Theia Data Center[??]. Each tile is displayed with its name in the Sentinel-2 nomenclature.(background map © OpenStreetMap contributors)

TABLE II: Number of pixels for each data set

Training	*Validation*	*Test*
92 000	23 000	230 000

ones described in [5]. Indeed, surface reflectance time-series and cloud/shadow masks were produced using the MAJA pre-processing chain [8]. All the bands at 20m/pixel were spatially up-sampled to 10m/pixel using the Orfeo Toolbox [9]. 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 [6] 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 between the 27 tiles results in $T = 303$ dates. Besides, the spectral dimension D corresponds to the 10 spectral bands added the three spectral indices (NDVI, NDWI, brightness) i.e. $D = 13$.

The reference data used in this work is composed of $C = 23$ land cover classes ranging from artificial areas to vegetation and water bodies. The nomenclature of the 23 land cover classes can be found in Table III. A detailed description of the data set can be found in [5]. 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 the Table II. To correctly estimate the classification accuracy, 9 runs with different random pixel samplings were done.

Standardization was performed for the valid acquisitions dates. Mean and standard deviation were estimated for each spectral band and for each spectral indice on the *training* data set and then used to standardize the others data sets (*training*, *validation*, *test*) [10].

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

B. Competitive methods

As described in [5], the SVGP model had great 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 features. By using end-to-end training, as described in Section III, 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 [5]
- Lightweight Temporal Self-Attention (LTAE) described in [11].

The mTAN $h_{\boldsymbol{\theta}_1}$ was described in the previous sections II-C and III. 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. 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. Therefore, the LTAE classifier was also studied without the mTAN and this method is called *raw-LTAE*.

The optimizer parameters for each model were found by trial and error and are described in Table VIII in Appendix B.

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. Then, the representation obtained from the mTAN is studied as well as the versability of its similarity kernel.

A. Performance results

In the following, the classification metrics were computed using the *test* data set (27 tiles) over the 9 runs of each model trained on the *training* data set.

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. Moreover, the number of latent spectral features was equal to the number of spectral features such as $D' = D = 13$. The representation $\mathbf{Z}(\mathbf{R})$ obtained using the mTAN is described by $d = R \times D' = 481$ spectro-temporal features.

The *mTAN-SVGP* model can be easily compared with the *Gapfilled-GP* model: a simple SVGP classifier that take as input linearly interpolated data. The data are linearly resampled onto a common set of latent dates with the same interval (10 days) and the same total number of dates (37 dates). Therefore, with the *Gapfilled-GP* model, the same number of spectro-temporal features are defined.

The representation obtained with the mTAN seems to give more advantages to the SVGP classifier than the linearly interpolated data. Indeed, as shown in Table IV, the *mTAN-SVGP* model is 10 points above the *Gapfilled-GP* model. To have a comparison, the linearly interpolated data was also feed into a simple Random Forest (RF) classifier with 100 trees called *Gapfilled-RF*. As found in [5], the overall accuracy with the *Gapfilled-GP* model is two points above than with the *Gapfilled-RF*.

TABLE IV: Averaged overall accuracy (OA) for the *mTAN-SVGP*, *Gapfilled-GP* and *Gapfilled-RF* models (mean % $\pm$ standard deviation computed with 9 runs)

mTAN-SVGP	*Gapfilled-GP*	*Gapfilled-RF*
77.44 $\pm$ 0.15	67.25 $\pm$ 0.37	65.37 $\pm$ 0.43

2) Spectral and temporal reduction: As described in Section III-C, the estimation of parameters $\boldsymbol{\theta}_2$ of the SVGP is highly dependent on the number of spectro-temporal features $d = R \times D'$ with the following term: $M \times d$. A high number of parameters is time-consuming and reducing the number of features d could be beneficial for the convergence of the algorithm. As shown in Fig. 3, reducing the number of latent dates from $R = 37$ to $R = 13$ and the number of latent spectral features from $D' = 13$ to $D' = 9$ in the mTAN has a negligible effect on the overall accuracy (i.e. from 77.44 to 77.23). However, the number of parameters $\boldsymbol{\theta}_2$ is almost reduced by a factor of 4 (i.e. from 584 200 to 155 250 parameters) and the training times are divided by two, as described in Fig. 4.

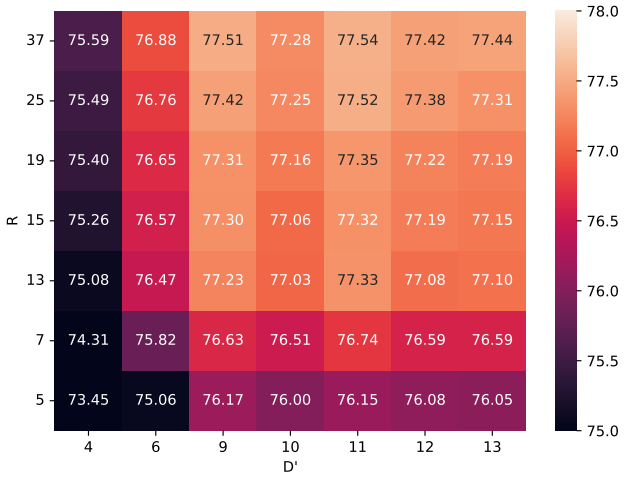


Fig. 3: Averaged overall accuracy (OA) for $H = 1$ (mean in % computed over 9 different runs) with R the number of latent dates and D' the number of latent spectral features.

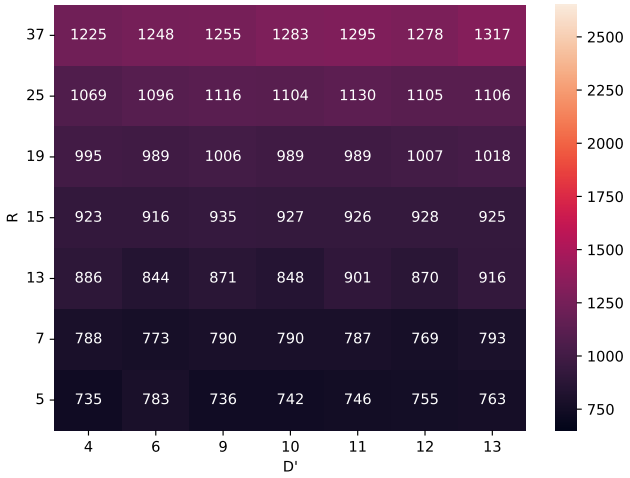


Fig. 4: Averaged training times for $H = 1$ (mean in sec computed over 9 different runs) with R the number of latent dates and D' the number of latent spectral features.

By changing R the number of latent dates in the temporal grid, the positions of these latent dates are modified because the temporal grid is defined with a regular interval. Thus, modifying the positions of the latent dates do not have any influence on the overall accuracy on the condition that there are enough latent dates R.

In addition, the number of time embedding functions H has little impact on the classification performances as shown in Appendix C in Fig. 10. Besides, from $H = 1$ to $H = 3$, the training time can be increased by a factor of 2 as shown in Appendix C in Fig. 11.

Next, we will focus on the *mTAN-SVGP* model with $R = 13$ latent dates, $D' = 9$ latent spectral features and $H = 1$ time embedding function.

3) Spatial positional encoding: The spatial information used to compute the positional encoded matrix $\mathbf{P}_{\psi_1,\psi_2}$ is composed of the longitude ψ_1 and the latitude ψ_2 in meters in the Lambert 93 projection. Two different activation functions are used in the two layers perceptron, the first one is a ReLU and the second one is a Sigmoid in order to bound the weights between zero and one. 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.

TABLE V: Averaged overall accuracy (OA) without (standard) and with the spatial positional encoded matrix (with $\mathbf{P}_{\psi_1,\psi_2}$) in the mTAN for the model *mTAN-SVGP* (mean % $\pm$ standard deviation computed with 9 runs)

	Standard	With $\mathbf{P}_{\psi_1,\psi_2}$
$R = 13$ $D' = 9$	77.23 $\pm$ 0.17	78.63 $\pm$ 0.16

Next, we will focus on the *mTAN-SVGP* model with $R = 13$ latent dates, $D' = 9$ latent spectral features, $H = 1$ time embedding function and the spatial positional encoded matrix $\mathbf{P}_{\psi_1,\psi_2}$.

B. Representation obtained from the mTAN

The representation $\hat{\mathbf{x}}_j$ obtained from the mTAN from (7) with the *mTAN-SVGP* model is totally used for the classification. It is not a signal reconstruction i.e. there is no specific term in the loss. However, as it is a projection onto a regular grid, it can be easily compared to the raw data and to the gapfilled data i.e. linearly interpolated data. Indeed, the spectral feature j is not modified because linear mixing has not yet been applied.

Fig. 5 represents the comparison of these three spectral profils (raw, gapfilled and mTAN) for one pixel labeled as "CORN". The spectral feature corresponds to the NDVI and the all year 2018 is considerated. The mTAN representation was obtained from the *mTAN-SVGP* model trained with a total of $R = 13$ latent dates with an interval time of 30 days, $D' = 9$ latent spectral features, $H = 1$ time embedding function and the spatial positional encoded matrix $\mathbf{P}_{\psi_1,\psi_2}$. The mTAN representation obtained in Fig. 5 is similar to the raw data or to the gapfilled data even if it is not a signal reconstruction.

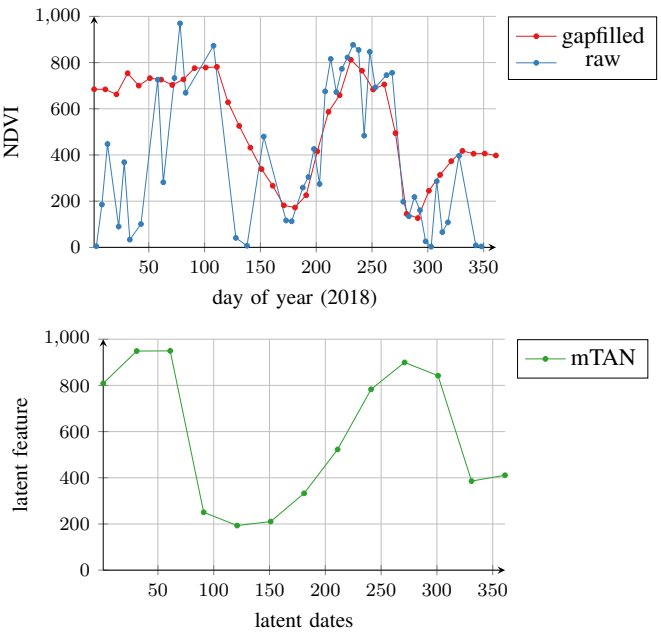
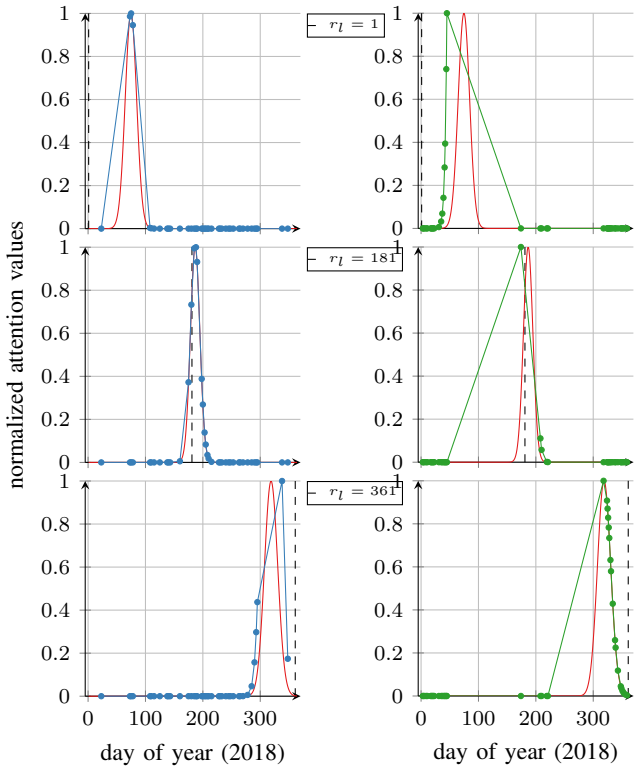


Fig. 5: Spectral profils for a pixel labeled "CORN". The spectral feature corresponds to the NDVI, its value is converting in integer using a factor of 1000.
– raw corresponds to the raw data, the outlier values have been removed in order to have a comprehensive plot.
– gapfilled corresponds to the value 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 ($D' = 9$).

C. Versability of the similarity kernel

As defined in Section II-C, by using *attention* and *embedding* mechanisms, the similarity kernel is adaptative to our problem: the classification task. The versability of the similarity kernel can be shown by computing the attention value $\boldsymbol{\gamma}_r$ defined in (6) for different latent dates r_l and for different set of observed dates $\mathbf{T}$. Fig. (6) represents the normalized attention values $\boldsymbol{\gamma}_r^n$ computed for three different latent dates: $r_l \in \{1, 181, 361\}$ and with three different set of observed dates $\mathbf{T}$. The first one, in red in both figures, is equal to $\mathbf{T} = \{1, .., 365\}$ with a regular interval of $\tau = 1$ day. The last two, respectively in blue and green in Fig. (6a) and (6b), are $\mathbf{T} = \mathbf{T}^i$ and $\mathbf{T} = \mathbf{T}^{i'}$ with i and i' two random pixels.

The kernel can be described as heteroscedastic: for different r_l latent dates, the bandwidth of our kernel varies. As shown in Fig. (6), 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 $r_l = 181$. In this case, the kernel takes the information further. Moreover, the kernel can be qualified as asymmetric for both the ith and i'th random pixels as shown in Fig. (6a) and (6b). To conclude, the kernel adapts itself according to the latent date r_l and to the set of observed dates $\mathbf{T}$.



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

Fig. 6: Normalized attention values γ_r^n for three different latent dates: $r_l \in \{1, 181, 361\}$. — corresponds to γ_r^n computed with $\mathbf{T} = \{1, .., 365\}$ with a regular interval of $\tau = 1$ day. — and — correspond to γ_r^n respectively computed with $\mathbf{T} = \mathbf{T}^i$ for the ith random pixel and $\mathbf{T} = \mathbf{T}^{i'}$ for the i'th random pixel.

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: both quantitative and qualitative. Then, a further comparative study is made between the *mTAN-SVGP* and the *raw-LTAE*.

A. Performance results

The *mTAN-SVGP* model is compared with 3 different models: *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*. In the following sections, as described in the previous section, the mTAN is defined with $R = 13$ latent dates, $D' = 9$ latent spectral features, $H = 1$ time embedding function and the use of the spatial positional encoding matrix $\mathbf{P}_{\psi_1,\psi_2}$. The *raw-LTAE* model is the only one not using the mTAN. Thus, no spectral or temporal reduction was implemented in this model. However, the spatial positional encoding matrix $\mathbf{P}_{\psi_1,\psi_2}$ was implemented in this model.

1) Quantitative results: As in Section V-A, the classification metrics were computed using the *test* data set (27 tiles) over the 9 runs of each model trained on the *training* data set.

As shown in Fig. 7, 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 three points below the *raw-LTAE* model.

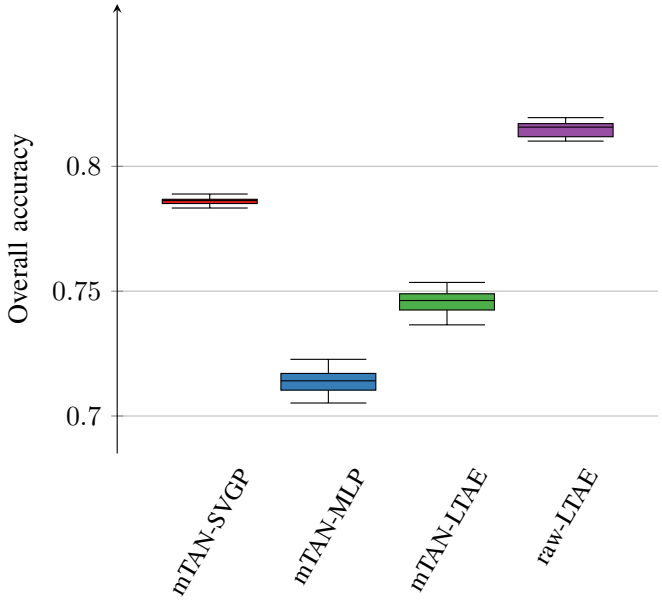


Fig. 7: Boxplots of the overall accuracy for each model (*mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*) computed over 9 runs.

However, the training time of the *mTAN-SVGP* model is 1.5 times shorter than the *raw-LTAE* as shown in Table VI. The averaged training times were computed over 9 runs with one NVIDIA Tesla V100 GPU.

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

mTAN-SVGP	*mTAN-MLP*	*mTAN-LTAE*	*raw-LTAE*
834	1207	840	1279

The number of trainable parameters for each method is summarized in Table VII. By using a spectro-temporal reduction with the mTAN, the number of parameters is considerably reduced. For example, for the LTAE, this number is reduced by four. However, as shown in Fig. 7, the overall accuracy of the *mTAN-LTAE* model is seven points below the *raw-LTAE* model. On the other hand, the number of trainable parameters for *mTAN-SVGP* is almost four times smaller than *raw-LTAE* but its OA only three points below.

TABLE VII: Number of trainable parameters for the models: *mTAN-SVGP*, *mTAN-MLP*, *mTAN-LTAE* and *raw-LTAE*

mTAN-SVGP	*mTAN-MLP*	*mTAN-LTAE*	*raw-LTAE*
202 646	33 113	184 376	761 380

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 [12] on two different tiles: $T31TCJ$ and $T31TDJ$. 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. (8a) and (8b). The results obtained on this agricultural area on the $T31TCJ$ tile show that the pixels are more homogeneous (with less salt and pepper classification noise [13]) with the *mTAN-SVGP* compared to the *raw-LTAE*. All the land cover maps generated are available for download.

B. Robustness to the temporal sampling

The *raw-LTAE* showed great classification performances both in qualitative and quantitative results. 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 $\mathbf{T} = \{t_1, ..., t_T\}$ used for training. It is not the case of the classifier using the mTAN. Thus, the *mTAN-SVGP* can be described as frugal in constrast to the *raw-LTAE* (i.e. it achieves the same results for less energy).

An other interesting feature is the robustness to the temporal sampling. Indeed, he acquisition dates between pixels on two adjacent tiles are shifted by δ' days. For a pixel on a specific tile, a shift δ with $\delta \leq \delta'$ in the acquisition dates $\mathbf{T}$, should not impact the classification performance. To check the robustness to this temporal sampling, we used two different models: *mTAN-SVGP* and *raw-LTAE* both trained on the 27 tiles. The overall accuracy was computed with the *test* data set only on $T31TCJ$ tile. The acquisition dates $\mathbf{T}$ for the *test* data set were artificially shifted with different values: $\delta = \{0, 1, 2, 3, 5\}$ days. As shown in Fig. 9, the overall accuracy of the *mTAN-SVGP* model is not affected by this temporal shift δ. However, the overall accuracy of the *raw-LTAE* model is greatly impacted by the temporal shift δ and is almost divided by 1.5 with $\delta = 5$ days. By using a linear smoothing, the *mTAN-SVGP* model is more robust to the temporal sampling than the *raw-LTAE* model which only use temporal attention mecanisms.

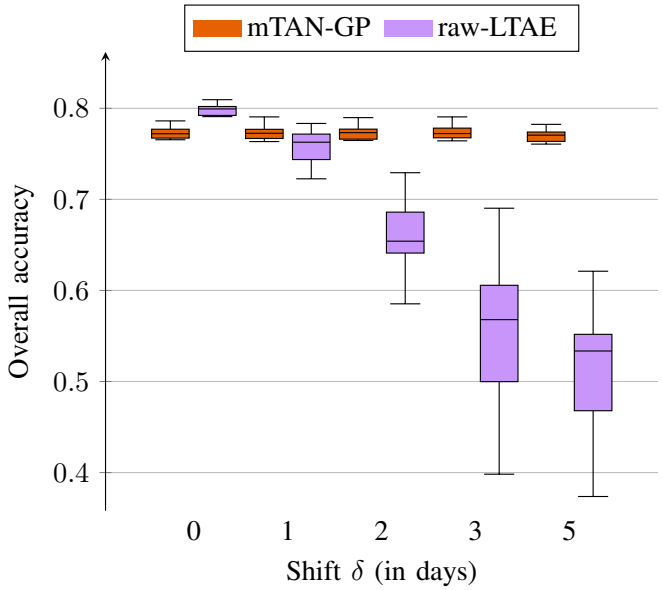
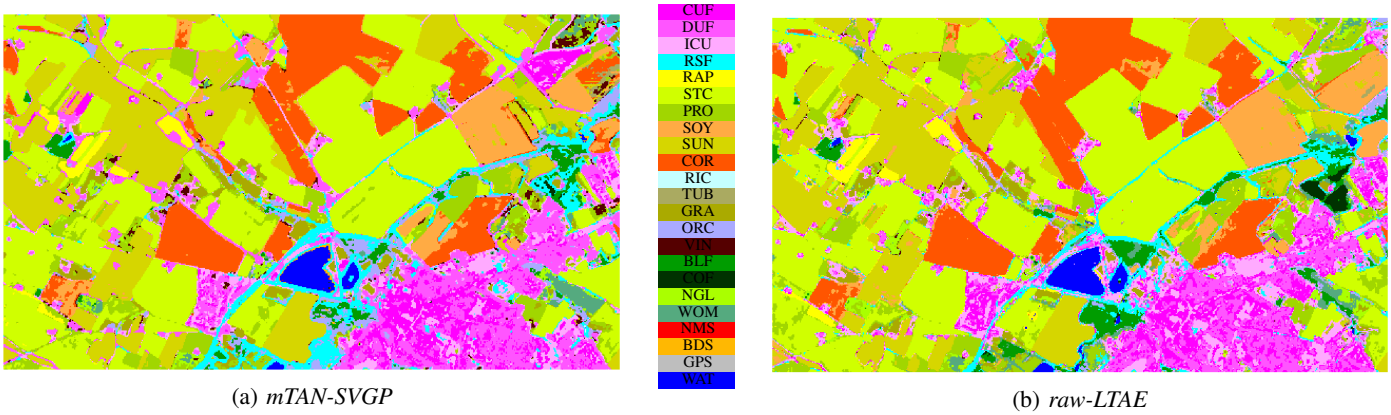


Fig. 9: Boxplots of the overall accuracy for the *mTAN-SVGP* and *raw-LTAE* models computed with the *test* data set only on the $T31TCJ$ tile over 9 runs. The models were trained and validated on the all 27 tiles. The acquisition dates $\mathbf{T}$ for the test data set were artificially shifted with different values: $\delta = \{0, 1, 2, 3, 5\}$ days.

VII. Conclusions and perspectives

Acknowledgment

The authors would like to thank Julien Michel for the implementation of the mTAN. The authors would also 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.



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

Fig. 8: Land cover maps obtained on an agricultural area on the tile $T31TCJ$

REFERENCES

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC, 1 ed., July 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] J. Inglada, A. Vincent, M. Arias, and B. Tardy, *iota2-a25386*, July 2016. https://doi.org/10.5281/zenodo.58150.

[13] 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.

[14] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Oct. 2008. Version 20081110.

APPENDIX A
SPATIAL POSITIONAL ENCODING

$$k(\mathbf{Z}^i, \mathbf{Z}^{i'}) = \alpha \exp\left(-\frac{\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_2^2}{2\sigma^2}\right)$$
$$= \alpha\left(k^{\lambda t}(\mathbf{x}^i, \mathbf{x}^{i'}) \times k^{\psi}(\{\psi_1^i, \psi_2^i\}, \{\psi_1^{i'}, \psi_2^{i'}\}) \times k^{\lambda t \psi}(\mathbf{x}^i\{\psi_1^i, \psi_2^i\}, \mathbf{x}^{i'}\{\psi_1^{i'}, \psi_2^{i'}\})\right)$$

with

$$\begin{aligned}
\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_2^2 &= \|\text{vec}(\mathbf{Z}^i - \mathbf{Z}^{i'})\|_2^2 \\
&= \|\text{vec}(\mathbf{Z}^i) - \text{vec}(\mathbf{Z}^{i'})\|_2^2 \\
&= \|\text{vec}(\mathbf{B}\tilde{\mathbf{X}}^{i*}\mathbf{\Gamma}^i) - \text{vec}(\mathbf{B}\tilde{\mathbf{X}}^{i'*}\mathbf{\Gamma}^{i'})\|_2^2 \\
&= \|\text{vec}(\mathbf{B}(\mathbf{X}^{i*} + \mathbf{P}^i)\mathbf{\Gamma}^i) - \text{vec}(\mathbf{B}(\mathbf{X}^{i'*} + \mathbf{P}^{i'})\mathbf{\Gamma}^{i'})\|_2^2 \\
&= \|((\mathbf{\Gamma}^i)^\top \otimes \mathbf{B})\text{vec}(\mathbf{X}^{i*} + \mathbf{P}^i) - ((\mathbf{\Gamma}^{i'})^\top \otimes \mathbf{B})\text{vec}(\mathbf{X}^{i'*} + \mathbf{P}^{i'})\|_2^2 \text{ from [14, Chapter 10]} \\
&= \|(\Omega^i\text{vec}(\mathbf{X}^{i*} + \mathbf{P}^i) - \Omega^{i'}\text{vec}(\mathbf{X}^{i'*} + \mathbf{P}^{i'})\|_2^2 \\
&= \|(\Omega^i\text{vec}(\mathbf{X}^{i*}) - \Omega^{i'}\text{vec}(\mathbf{X}^{i'*})) - (\Omega^i\text{vec}(\mathbf{P}^i) - \Omega^{i'}\text{vec}(\mathbf{P}^{i'}))\|_2^2 \\
&= \|\Omega^i\text{vec}(\mathbf{X}^{i*}) - \Omega^{i'}\text{vec}(\mathbf{X}^{i'*})\|_2^2 + \|\Omega^i\text{vec}(\mathbf{P}^i) - \Omega^{i'}\text{vec}(\mathbf{P}^{i'})\|_2^2 \\
&\quad - 2[\Omega^i\text{vec}(\mathbf{X}^{i*}) - \Omega^{i'}\text{vec}(\mathbf{X}^{i'*})]^\top[\Omega^i\text{vec}(\mathbf{P}^i) - \Omega^{i'}\text{vec}(\mathbf{P}^{i'})]
\end{aligned}$$

$$\begin{aligned}
\|\mathbf{Z}^i - \mathbf{Z}^{i'}\|_2^2 &= \|\mathbf{Z}^i\|_2^2 + \|\mathbf{Z}^{i'}\|_2^2 - 2\langle\mathbf{Z}^i, \mathbf{Z}^{i'}\rangle_2 \\
&= \|\mathbf{B}(\mathbf{X}^{i*} + \mathbf{P}^i)\mathbf{\Gamma}^i\|_2^2 + \|\mathbf{B}(\mathbf{X}^{i'*} + \mathbf{P}^{i'})\mathbf{\Gamma}^{i'}\|_2^2 - 2\langle\mathbf{Z}^i, \mathbf{Z}^{i'}\rangle_2 \\
&= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i\|_2^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\|_2^2 + 2\langle\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\rangle_2 \\
&\quad + \|\mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_2^2 + \|\mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_2^2 + 2\langle\mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_2 \\
&\quad - 2\left(\langle\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_2 + \langle\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_2 + \langle\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_2 + \langle\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_2\right) \\
&= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_2^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_2^2 \\
&\quad + 2\left(\langle\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i\rangle_2 + \langle\mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_2 - \langle\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i, \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\rangle_2 - \langle\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i, \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\rangle_2\right) \\
&= \|\mathbf{B}\mathbf{X}^{i*}\mathbf{\Gamma}^i - \mathbf{B}\mathbf{X}^{i'*}\mathbf{\Gamma}^{i'}\|_2^2 + \|\mathbf{B}\mathbf{P}^i\mathbf{\Gamma}^i - \mathbf{B}\mathbf{P}^{i'}\mathbf{\Gamma}^{i'}\|_2^2 + 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_2
\end{aligned}$$

and with

$$k^{\lambda t}(\mathbf{x}^i, \mathbf{x}^{i'}) = \exp\left(-\frac{\|\Omega^i\text{vec}(\mathbf{X}^{i*}) - \Omega^{i'}\text{vec}(\mathbf{X}^{i'*})\|_2^2}{2\sigma^2}\right)$$

$$k^{\psi}(\{\psi_1^i, \psi_2^i\}, \{\psi_1^{i'}, \psi_2^{i'}\}) = \exp\left(-\frac{\|\Omega^i\text{vec}(\text{MLP}(\varphi(\psi_1^i, \psi_2^i)) - \Omega^{i'}\text{vec}(\text{MLP}(\varphi(\psi_1^{i'}, \psi_2^{i'}))\|_2^2}{2\sigma^2}\right)$$

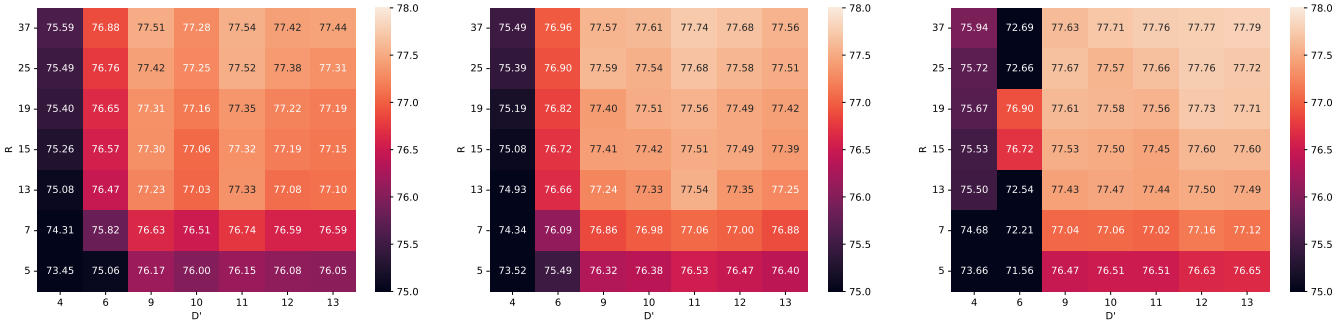
$$k^{\lambda t \psi}(\mathbf{x}^i\{\psi_1^i, \psi_2^i\}, \mathbf{x}^{i'}\{\psi_1^{i'}, \psi_2^{i'}\}) = -2 \times \exp\left(-\frac{???}{2\sigma^2}\right)$$

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

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

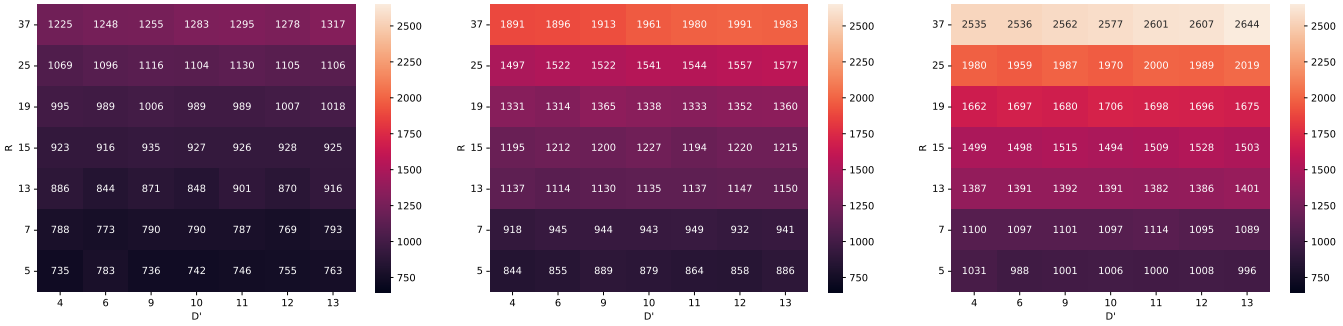
APPENDIX B
SOLVER PARAMETERS FOR EACH MODEL

APPENDIX C
INFLUENCE OF THE SPECTRAL AND TEMPORAL REDUCTION FOR DIFFERENT H TIME EMBEDDINGS



$H = 1$ $H = 2$ $H = 3$

Fig. 10: 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 time embeddings.



$H = 1$ $H = 2$ $H = 3$

Fig. 11: 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 time embeddings.