Forecasting water resources from satellite image time series using a graph-based learning strategy

Corentin Dufourg[1], Charlotte Pelletier[1], Stéphane May[2], Sébastien Lefèvre[1]

[1] Université Bretagne Sud, IRISA, UMR CNRS 6074, Vannes, France
(corentin.dufourg, charlotte.pelletier, sebastien.lefevre)@univ-ubs.fr
[2] Centre National d’Études Spatiales (CNES), Toulouse, France - stephane.may@cnes.fr

Keywords: Satellite Image Time Series, GraphCast, Forecasting, SEN2DWATER, Normalized Difference Water Index.

Abstract

In the context of climate change, it is important to monitor the dynamics of the Earth’s surface in order to prevent extreme weather phenomena such as floods and droughts. To this end, global meteorological forecasting is constantly being improved, with a recent breakthrough in deep learning methods. In this paper, we propose to adapt a recent weather forecasting architecture, called GraphCast, to a water resources forecasting task using high-resolution satellite image time series (SITS). Based on an intermediate mesh, the data geometry used within the network is adapted to match high spatial resolution data acquired in two-dimensional space. In particular, we introduce a predefined irregular mesh based on a segmentation map to guide the network’s predictions and bring more detail to specific areas. We conduct experiments to forecast water resources index two months ahead on lakes and rivers in Italy and Spain. We demonstrate that our adaptation of GraphCast outperforms the existing frameworks designed for SITS analysis. It also showed stable results for the main hyperparameter, *i.e.*, the number of superpixels. We conclude that adapting global meteorological forecasting methods to SITS settings can be beneficial for high spatial resolution predictions.

1. Introduction

The increase in the number of sensors onboard satellites has created a vast amount of data that can help monitor the Earth. In particular, the high revisit rate offered by certain satellite constellations makes it possible to track the dynamics of the Earth’s surface. For example, the two Sentinel-2 satellites acquire images every five days at the equator with a spatial resolution of 10 meters at best. The stack of satellite images covering the same area at different times is known as satellite image time series (SITS). These data serve as valuable resources for global monitoring, including land cover land use mapping, deforestation monitoring, landslide detection, natural resources management and various other applications.

Over the past decade, significant advances have been achieved in the automatic processing of SITS, notably through the adoption of deep learning techniques (Miller et al., 2024), demonstrated for example in the BreizhCrops benchmark (Rußwurm et al., 2020). SITS form complex data cubes structured by their spatial and temporal dimensions, requiring the development of specific architectures. In the context of SITS semantic segmentation (*i.e.*, one prediction for each time-series pixel), recent strategies developed architectures that make the most of both dimensions through dual-branch architectures (Interdonato et al., 2019), ConvLSTM (Rußwurm and Körner, 2018), U-Net with temporal attention encoder (TAE) (Sainte Fare Garnot and Landrieu, 2021), or variants of Vision Transformers (Tarasiou et al., 2023, Voelsen et al., 2023). Similar approaches can be also exploited for extrinsic regression, *e.g.*, yield estimation (Sun et al., 2020), and forecasting tasks (Moskolaï et al., 2021).

Another investigated approach for SITS analysis relies on graphs, a well-established mathematical tool for data lying in non-Euclidean domains, such as social networks, chemical molecules, or point clouds. Recent advances in graph-based learning, fostered by graph neural networks (GNNs), also offer a compelling alternative to grid-based deep learning techniques for certain image-related tasks (Jiao et al., 2022) such as scene understanding. GNNs leverage prior knowledge about graph structures to effectively model complex relationships and interactions between image pixels or regions by processing them as nodes in a graph. This allows GNNs to capture structural information in images, making them suitable for tasks where context understanding and relational reasoning are crucial. In addition, a graph, used as an input of GNNs, can handle data of variable sizes, which is a significant advantage for SITS that have irregular temporal sampling and whose pixels may be saturated or perturbed by clouds and their shadows, leading to a disruption in the spatial regularity of the images.

The interest in graph-based learning for spatio-temporal data exploded over the past five years in environmental applications, and more specifically for global estimation of meteorological conditions such as weather (Lin et al., 2022, Keisler, 2022) or sea surface temperature (Cachay et al., 2021, Ning et al., 2024), framed as forecasting tasks. Recently, GraphCast (Lam et al., 2023) created a breakthrough in environmental applications by releasing a graph-based model that can predict global weather conditions up to 10 days in advance at a 6-hour resolution and 0.25 degrees. We note that most of these approaches are not based on SITS but on derived spatio-temporal datasets such as ERA5, which combines several data sources through data assimilation. In the context of SITS data of a medium or high spatial resolution (below 50 meters), the main application is land cover mapping, for which GNNs have also proven to be efficient (Kavran et al., 2023, Tulczyjew et al., 2022, Censi et al., 2021). For example, our prior work (Dufourg et al., 2023) demonstrated the potential of graph neural networks (GNNs) for dense land cover predictions from SITS (*i.e.*, predictions for all the pixels and all timestamps), against Random Forests and Multi-Layer Perceptron classifiers. The graph was obtained through a segmentation step, where each node represents a segment, referred to as an object. The experiments highlighted the

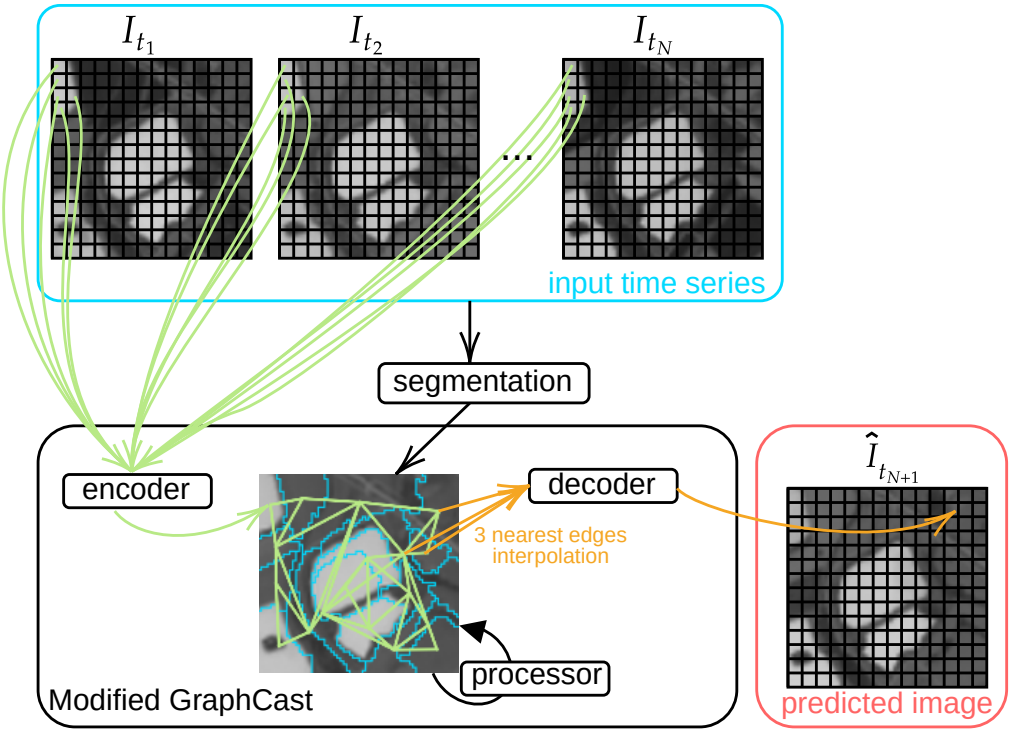

Figure 1. Overview of our proposed approach based on GraphCast. The input is a sequence of satellite image time series acquired at date t_1, t_2, ..., t_N. The output is the predicted image at t_{N+1}.

importance of integrating each object’s neighborhood feature with the inner feature to understand the object’s context and improve local representation.

Inspired by GraphCast, we propose to further bridge the gap between graph-based learning and SITS analysis by focusing on a forecasting task from high spatial resolution SITS. The objective is to estimate water resources across different basins. Specifically, the experiments are conducted on the SEN2DWATER dataset (Mauro et al., 2023). GraphCast is adapted to predict the next image of a sequence of satellite images.

The paper is organized as follows: Section 2 presents the proposed approach together with some background on GraphCast. Section 3 describes the dataset and experimental settings, while Section 4 reports the results. Finally, we draw conclusions in Section 5.

2. Methodology

In this section, we detail our forecasting pipeline, whose objective is to predict the water basin resources at a given date using a sequence of satellite images. Since our method heavily depends on GraphCast, we briefly introduce it before discussing our approach in detail.

2.1 Background on GraphCast

In a nutshell, GraphCast (Lam et al., 2023), implemented by Google DeepMind, is an autoregressive graph-based model that predicts global weather conditions at a 6-hour resolution. Following the trend of GNN-based learned simulators (Pfaff et al., 2020, Keisler, 2022), the architecture consists of an encoder, a processor and a decoder. The model is fed with two consecutive weather states, each represented by six climatic variables from 37 atmospheric levels, five surface variables, and some forcing states, including time and spatial information. The encoder projects the weather state of a local region (*i.e.*, a group of pixels) into the nodes of a multi-mesh graph that models the Earth’s geometry at a scale suitable for calculations. In practice, an icosahedron (*i.e.*, a polyhedron with 20 faces, which are equilateral triangles), recognized for its ability to approximate a sphere, is refined iteratively six times. The nodes of the finest resolution and all the edges across the different resolutions form the multi-mesh. Then, the processor learns latent representations of the multi-mesh nodes via message passing. Finally, the decoder maps the learned features to each pixel using only the three nearest multi-mesh nodes. It predicts the weather state forecast as a residual, *i.e.*, it predicts the difference with the most recent input. All three modules (encoder, processor, and decoder) depend on GNNs (Battaglia et al., 2018). Additional technical details, *e.g.*, autoregressive training and the corresponding loss, can be found in the published paper (Lam et al., 2023).

2.2 Our pipeline

The objective is to forecast the next image of a sequence of N satellite images. The employed pipeline is presented in Figure 1. The input is a sequence of N images $\{I_{t_i}\}_{i=1}^{N}$ acquired at date t_i of size $H \times W$. The output corresponds to the prediction for the next image in the series $\hat{I}_{t_{N+1}}$. The employed graph-based model follows the encoder-processor-decoder structure of GraphCast.

Unlike GraphCast, which aims to provide global predictions, our study focuses on location-specific images covering small areas. This has led to two major differences with GraphCast: (i) use of a single mesh as in (Keisler, 2022), and (ii) substitution of a global mesh by a region-specific mesh. Compared

to meteorological predictions that require capturing long-range dependencies, we hypothesize that a single mesh, *i.e.*, a single resolution, is sufficient to characterize local phenomena occurring in water basins.

Furthermore, GraphCast makes predictions at a global scale, employing a mesh adapted to approximate a sphere. In our study, the predictions are performed on a more limited spatial extent depicted at a higher spatial resolution, making the icosahedron inappropriate. Hence, we propose to rely on a segmentation map to create a region adjacency graph used as a predefined mesh. Using an irregular mesh, rather than a regular grid like the rectangular meshes used in CNN or Vision Transformers, or GraphCast's icosahedral mesh, can bring more detail to specific areas and limit the smoothing effect.

While the pipeline can employ any segmentation algorithm, we chose the simple linear iterative clustering (SLIC) algorithm (Achanta et al., 2012), which is fast and efficient. SLIC partitions images into regions of similar intensities using a k-means clustering algorithm specially designed for images. It has two hyperparameters: (i) the (approximate) number of superpixels that will be generated by the algorithm, and (ii) the compactness that controls the balance between spatial proximity and intensity similarity when forming superpixels, a high value leads to superpixels with a square shape. As the number of superpixels influences the typology of the mesh, we will study its impact in Section 4.2. A high value of the number of superpixels leads to many small superpixels, while a small value results in fewer and larger superpixels.

As a reminder, our objective is to derive a single mesh from the sequence of satellite images. While SLIC was originally developed for single natural RGB images, most implementations can process images with any arbitrary number of channels. In this study, two strategies to obtain a single mesh from multiple images are compared in Section 4.2: (i) apply SLIC to the last image of the sequence I_{t_N} assuming that the segmentation at the next time t_{N+1} is similar, and (ii) apply SLIC to the stack of images to consider dynamics.

A last modification concerns embedding of the node's features for the encoder. In GraphCast, the node's features are derived by feeding weather states concatenated together with spatial and temporal embeddings into a small multi-layer perceptron (MLP). The spatial embedding involves the computation of the sine of the latitude, and the sine and cosine of the longitude, while the temporal embedding is a normalization of the sine and cosine of the local time of day and the sine and cosine of the day of year. We adopt a similar strategy but encoded independently the variable (*i.e.*, estimation of the water resources content) to provide the model with a richer context. The spatial and temporal embeddings are also processed using a second independent MLP. Subsequently, the variable and spatio-temporal features are concatenated before being passed through a third MLP to obtain the node's features. In practice, we use the same spatial embedding but opt to use a sinusoidal positional encoding solely for the day of the year in the temporal embedding, as done in recent research on foundation models for SITS (Guo et al., 2023, Tseng et al., 2023).

3. Experimental Settings

3.1 Dataset

To carry out the experiments, we used the SEN2DWATER dataset (Mauro et al., 2023)[1]. The original dataset consists of 5 264 sequences of Sentinel-2 top-of-atmosphere image patches gathered in 17 different basins in Italia and Spain between July 2016 and December 2022. The areas cover the surroundings around lakes and rivers, mainly in agricultural regions where basin water resources might be used for irrigation. Each SITS patch has a size of 64 pixels $\times$ 64 pixels and a temporal resolution of around two months (only the less cloudy images were processed). We filtered out SITS that contained missing data in the original dataset, leading to a total of 3 682 sequences of Sentinel-2 image patches for the experiments.

The objective is to exploit past satellite images to forecast water resources, approximated by the Normalized Difference Water Index (NDWI). In practice, we computed the NDWI (McFeeters, 1996) using the green (B03) and near-infrared (B08) 10-meter resolution bands from Sentinel-2. This formulation makes it possible to detect water bodies and precise variations in water content, providing information regarding the overall turbidity (suspended sediments and chlorophyll α_t). Its sensitivity to the level of water makes it also a reasonable choice to forecast water resources in basins. However, NDWI might also be responsive to built-up content (*e.g.*, buildings, roads, and bridges) and atmospheric conditions, in particular clouds, leading to over-estimation of water bodies when applying a simple thresholding approach to identify water content (Xu, 2006).

To guarantee the independence of the train and test sets, we ensure that image patches from the same basin cannot be in different sets. The SITS patches are split using an 85:15 ratio. The training set is further split into train and validation sets, used to monitor the performance and decrease the learning rate during the model's optimization.

3.2 Implementation Details

The implementation of GraphCast is adapted from the original Python/JAX code[2] to a Pytorch implementation with the help of PyTorch Geometric (PyG) (Fey and Lenssen, 2019). We use the same GNN architectures for the encoder, processor, and decoders, but reduce the number of layers and units due to the smaller size of our dataset. In particular, our modified encoder consists of three MLPs used to embed the NDWI and the spatial and temporal information (see Section 2.2), followed by 1 graph convolutional layer with 64 units. The processor consists of 4 graph convolution layers, each of 64 units. The decoder is composed of 1 graph convolution layer of 64 units. All MLPs, including those used in graph convolutions, are composed of a fully connected layer, followed by a Hardswish activation function, a second fully connected layer, and finally a layer normalization. The final layer of our model has neither layer normalization nor activation function. As a recall, the model should predict residuals in the range $[-2, 2]$ as the NDWI is in the range $[-1, 1]$. Besides, the segmentation map used as the graph structure is generated with the SLIC algorithm implemented in

[1] Accessed on October 16, 2023. The dataset has been updated several times, notably to enrich it with Sentinel-1 radar images and additional water-bodies labels used to solve a semantic segmentation task (Russo et al., 2024).

[2] `https://github.com/google-deepmind/graphcast/`

	# Params	RMSE ↓	PSNR ↑	SSIM ↑	Runtime (min)
Input average	-	0.1550	23.32	0.7465	-
Persistence	-	0.1332	25.03	0.7897	-
LSTM	17,345	$0.1162_{\pm 0.0005}$	$25.53_{\pm 0.05}$	$\mathbf{0.8282}_{\pm 0.0005}$	22
ConvLSTM	150,721	$0.1197_{\pm 0.0029}$	$25.28_{\pm 0.19}$	$0.8113_{\pm 0.0030}$	26
TDCNN-ConvLSTM	407,681	$0.1111_{\pm 0.0008}$	$25.68_{\pm 0.08}$	$0.8083_{\pm 0.0008}$	46
Ours	228,673	$\mathbf{0.1097}_{\pm 0.0035}$	$\mathbf{26.42}_{\pm 0.27}$	$0.8170_{\pm 0.0070}$	41

Table 1. Number of parameters, RMSE, PSNR, SSIM and runtime for baseline models and our GraphCast adaptation. The mean and standard deviation are reported for three initializations of the model’s parameters. Bold values display the best performance for each metric.

Scikit-Image (van der Walt et al., 2014). By default, SLIC is applied to the last image of the input sequence with a number of superpixels of 256 and a compactness value of 0.1 The influence of this segmentation step will be studied in Section 4.2.

The proposed model is compared with three competitors proven to be efficient for SITS analysis (Rußwurm and Körner, 2017, Rußwurm and Körner, 2018, Mauro et al., 2023): (i) Long Short Term Memory (LSTM), (ii) ConvLSTM, and (iii) Time Distributed CNN ConvLSTM (TDCNN-ConvLSTM), the best-performing strategy (originally called TD-CNN) tested on the SEN2DWATER dataset (Mauro et al., 2023). The TDCNN-ConvLSTM architecture is similar to ConvLSTM but first performs an embedding of each image individually with 2D spatial convolutions, enriching the image’s context. While SITS patches are inputted in ConvLSTM and TDCNN-ConvLSTM, LSTM is applied pixel-wise and thus is agnostic to the spatial dimension. These three models predict directly the next image in the sequence, not the residuals as in GraphCast. We implemented and evaluated these three models on the curated dataset presented in Section 3.1. In short, the LSTM model consists of 1 LSTM cell of 64 units. Similarly, the ConvLSTM has 1 ConvLSTM layer of 64 units, with a convolutional kernel size of 3×3. The TDCNN-ConvLSTM has a similar structure but first embeds individually the images with a CNN composed of 4 layers, each of 64 units. For the three models, the last activation function is a hyperbolic tangent to yield prediction in the range $[-1, 1]$ adapted to NDWI. The implementation mirrors the work of Mauro et al. (2023). Implementation details are available at `https://github.com/corentin-dfg/graph4sen2dwater`.

Our revised version of GraphCast and the competitor models are also compared with two weak baselines: (i) input average that corresponds to predict the average image, *i.e.*, $\hat{I}_{t_{N+1}} = \frac{1}{N}\sum_{k=1}^{N} I_{t_k}$, and (ii) persistence that corresponds to predict the same image than the one observed at the latest timestamp, *i.e.*, $\hat{I}_{t_{N+1}} = I_{t_N}$.

All the networks are trained using the Huber loss for 50 epochs and Adam optimizer with an initial learning rate of 0.0001, decreased when the validation loss does not reduce over the past five epochs. We set the number of images N used for the predictions to six, which corresponds to a period of approximately one year, following the setting proposed in (Mauro et al., 2023). The experiments run on a CPU 12th Gen Intel® Core™ i7-12800H × 20 and an NVIDIA RTX A1000.

3.3 Evaluation Metrics

To assess the performance of the forecasting models, we compute widely used quality measures averaged over all the test images. In particular, we use reconstruction and perception-based metrics, including root mean square error (RMSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure

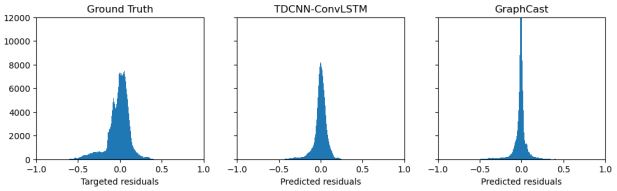

Figure 2. Distribution of the targeted NDWI residuals and the predicted ones by TDCNN-ConvLSTM and GraphCast for each pixel of the test set.

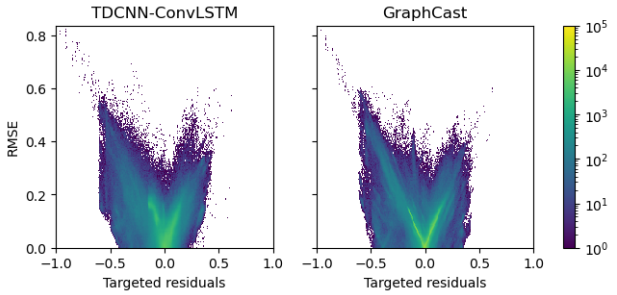

Figure 3. Scatter plot of the RMSE as a function of the targeted NDWI residuals for each pixel of the test set. A brighter color indicates a higher concentration of points.

(SSIM) (Wang et al., 2004). To assess the model’s stability, we repeat the experiments three times with different initializations of the network’s parameters and report the mean and standard deviation. Additionally, we will evaluate and compare visually the predicted NDWI images.

4. Experimental Results

4.1 Forecasting results in the SEN2DWATER dataset

First, our revised version of GraphCast (ours) is compared against the three competitors detailed in Section 3.2. Table 1 reports the RMSE, SSIM, and PSNR evaluation measures. The total runtime in minutes, summing training and inference times, is also provided. All competitor models outperform the two weak baselines, with GraphCast yielding the best performance on two evaluation metrics, demonstrating its potential beyond its initial global meteorological application. The LSTM obtains the lowest RMSE and PSNR results among the competitors, but it yields the best performance on SSIM, a perceptual metric, despite the independent processing of each pixel. To derive water content, we assume that a low reconstruction error is more critical than visually pleasant results. Besides, embedding individually the image patches in a higher dimensional space (TDCNN-ConvLSTM) helps decrease the reconstruction error of ConvLSTM. Finally, the runtime, about 40 minutes, is com-

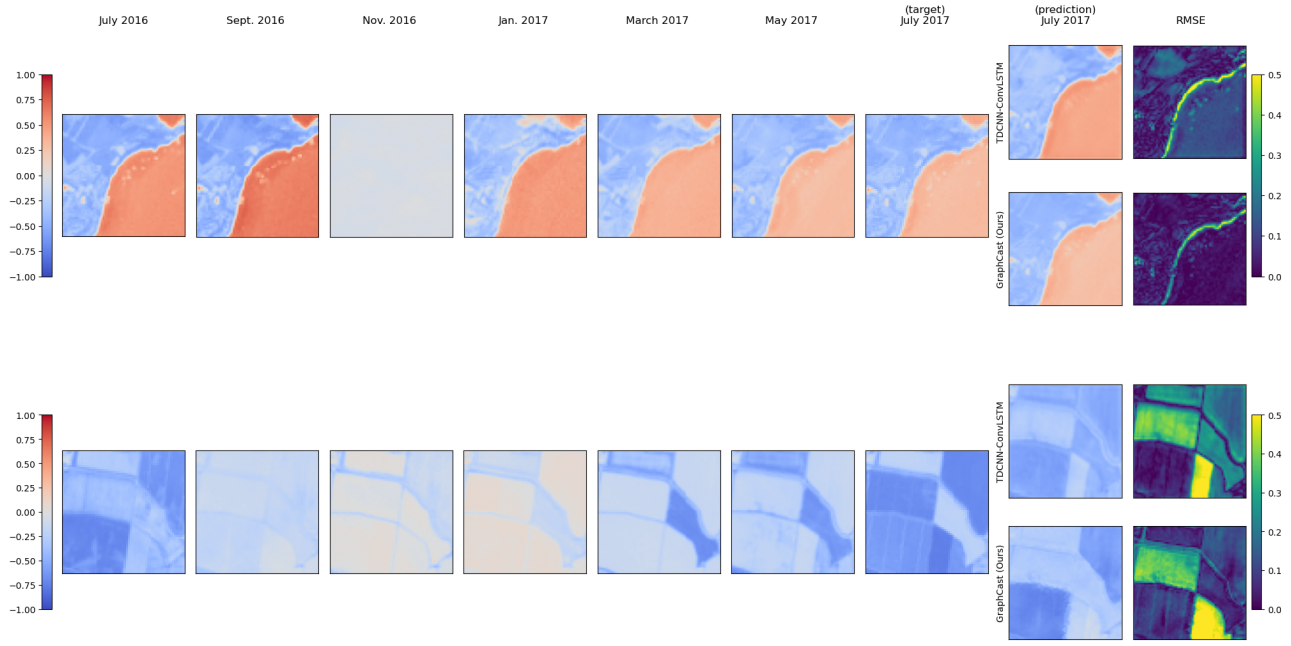

Figure 4. Prediction of the next SITS image by TDCNN-ConvLSTM and GraphCast. The model’s input images range from July 2016 to May 2017, to predict an acquisition in July 2017. *(top)* Lake Garda shores; despite data pre-processing, the November 2016 acquisition is cloudy. *(bottom)* Agricultural area on the banks of the Po River.

parable between TDCNN-ConvLSTM and our revised version of GraphCast.

In order to analyze the difference between our revised version of GraphCast and the best-performing competitor, TDCNN-ConvLSTM, Figure 2 displays the distribution of the residuals for the set of test images (*i.e.*,, the pixel-wise difference in NDWI per pixel between dates t_N and t_{N+1}) of these two methods compared with the expected distribution (ground truth). It is interesting to note that GraphCast tends to produce mainly small changes, surely linked to its residual approach. This remark is also in line with its original application, where the time steps between acquisitions are only a few hours. To further analyze the performance of the models on major changes, Figure 3 displays the RMSE of each pixel predicted with TDCNN-ConvLSTM and GraphCast as a function of the NDWI residue expected with the last image. A perfect prediction would give a horizontal line corresponding to zero error regardless of the expected change. However, Figure 3 shows a natural tendency of both models towards greater error when changes since the last acquisition are larger. In conclusion, GraphCast and TDCNN-ConvLSTM handle abrupt changes in the same way.

Finally, we visually assess the performance of the models. Figure 4 displays two examples NDWI time series and the predictions yielded by GraphCast (ours) and TDCNN-ConvLSTM, with corresponding error maps. It focuses on two types of areas representative of the SEN2DWATER dataset: lakeshores and farmland.

Concerning lakeshores, the area with a high NDWI indicates an open water surface, and the subtle variations at different dates are related to overall turbidity. On this SITS, we notice that the lake turbidity does not have a seasonal behavior as the NDWI in July 2017 is much closer to the last acquisition date in May 2017 than in July of the previous year. The forecasting of these non-seasonal dynamics was best captured by GraphCast. A portion of the image also depicts terrestrial vegetation, which seems to have a different dynamic to that of the lake. Water basins provide important irrigation resources, encouraging the development of surrounding crops. Although these often follow a seasonal pattern linked to plant growth, there is also a correlation with nearby water resources. It is therefore interesting to analyze an application of the proposed predictions to nearby crops.

As for the second SITS representing farmland, NDWI takes negative values for vegetation and zero values for bare soil. Although less suitable than the Normalized Difference Vegetation Index for analyzing vegetation, the NDWI can still be used to differentiate between bare and cultivated plots, and thus monitor seasonal harvesting dynamics. Focusing on the bottom SITS, the evolution of the fields follows a seasonal logic; the July 2017 acquisition is similar to that of July 2016 but shows a clear change compared to May 2017. This dynamic was learned by both TDCNN-ConvLSTM and GraphCast, as shown by the field prediction at the bottom left, and demonstrates the capability of both models to capture temporal dependencies. However, GraphCast seems to have better grasped these dependencies, given the predictions for the top-right fields, which nevertheless follow the same temporal pattern. Significant reconstruction errors occurred in two fields for both forecast models. They are due to a sudden change that cannot be predicted from the input data alone. This shows the importance of the length of the input time series for handling seasonal changes, which is studied in Section 4.3. It also demonstrates the inability of these models to extrapolate patterns they have never seen. Their application as they stand, for example, to predict non-seasonal extreme events such as floods or droughts, is therefore limited.

4.2 Influence of the segmentation step

In this section, we explore the influence of the segmentation step, in particular the number of superpixels. Figure 5 reports

the performance for an increasing number of segments ranging from 1 (each patch represents only one segment) to 4,096 segments (each pixel corresponds to one segment). The performance measured by the reconstruction error remains stable across a varying number of superpixels, as long as a minimal geometry is present, *i.e.*, with at least a dozen of superpixels. It shows that the dynamics observed in SITS can be processed at a higher level than the pixel (*i.e.*, the region) using a coarse intermediate geometric representation. The use of encoders and decoders to switch from fine geometry to mesh, and vice versa, also enables pixel-level precision to be maintained without compromising visual quality.

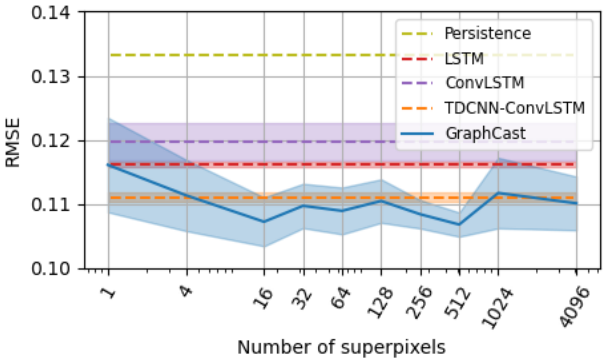

Figure 5. Influence of the number of superpixels used to create the mesh in our revised version of GraphCast. The mean and standard deviation are reported for three initializations of the model's parameters.

Further, Table 2 displays the performance for two segmentation strategies—SLIC applied to the stack of input images or SLIC applied to the last image of the sequence (default setting). For this experiment, the number of superpixels is set to 256. Although results show a slight performance gain when applying SLIC to the last image, the performance measures are within the same range. We argue that applying SLIC to the stack of input images might be beneficial if cloudy images are not curated as is the case in SEN2DWATER.

	RMSE ↓	PSNR ↑	SSIM ↑
Last-date SLIC	**0.1097**$_{\pm 0.0035}$	**26.42**$_{\pm 0.27}$	**0.8170**$_{\pm 0.0070}$
Multi-date SLIC	0.1101$_{\pm 0.0034}$	26.12$_{\pm 0.45}$	0.8151$_{\pm 0.0076}$

Table 2. Influence of the segmentation strategy. The mean and standard deviation are reported for three initializations of the model's parameters. Bold values display the best performance for each metric.

4.3 Influence of the length of the time series

The impact of the length of the input time series (N) is evaluated for our revised version of GraphCast. In particular, we assess the performance for variable input time series lengths ranging from 1 to 6. As there is no causal mechanism in the model contrary to recurrent architectures, the model has to learn by itself the temporal dependencies between the features. Having more acquisition dates as input should enable it to find these dependencies more easily. Indeed, according to Table 3, the longer the series, the less error the model makes. As expected, the model fed with the highest number of dates ($N = 6$) yields the best performance. It is important to note that when N is equal to 6, the model views a time series over one year. Hence, the image to be predicted, around July-September, was acquired for a season on which the model had been trained. Interestingly, the models that exploit a single image ($N = 1$) or two images ($N = 2$) as in GraphCast underperform and exhibit performance below the weak persistence baseline. It underlines the need to feed the model with long time series representing variations across the year for this forecasting task.

# input dates	RMSE ↓	PSNR ↑	SSIM ↑
1	0.1409$_{\pm 0.0085}$	24.28$_{\pm 0.65}$	0.7593$_{\pm 0.0218}$
2	0.1455$_{\pm 0.0013}$	23.72$_{\pm 0.18}$	0.7294$_{\pm 0.0098}$
3	0.1330$_{\pm 0.0157}$	24.75$_{\pm 1.28}$	0.7538$_{\pm 0.0421}$
4	0.1319$_{\pm 0.0020}$	24.77$_{\pm 0.09}$	0.7662$_{\pm 0.0092}$
5	0.1200$_{\pm 0.0034}$	25.42$_{\pm 0.40}$	0.7957$_{\pm 0.0082}$
6	**0.1097**$_{\pm 0.0035}$	**26.42**$_{\pm 0.27}$	**0.8170**$_{\pm 0.0070}$

Table 3. Forecasting results of our revised version of GraphCast for varying time series length N. The mean and standard deviation are reported for three initializations of the model's parameters. Bold values display the best performance for each metric.

4.4 Impact of the temporal and spatial encodings

In this section, we perform an ablation of the temporal and spatial encodings. Table 4 reports the results for four settings: no encoding, spatial encoding only, temporal encoding only, and both spatial and temporal encoding (our model). Temporal encoding slightly improves performance over no encoding, by providing additional information on acquisition dates, allowing the model to be guided towards seasonal dependencies. We also note that the simultaneous use of spatial and temporal encoding gives the best results. Surprisingly, adding spatial encoding alone performs lower than without any encoding. We were unable to explain this behaviour, which calls for further investigation and potentially better encoding of spatial information.

Spatial	Temporal	RMSE ↓	PSNR ↑	SSIM ↑
✗	✗	0.1150$_{\pm 0.0052}$	25.38$_{\pm 0.35}$	0.8123$_{\pm 0.0024}$
✓	✗	0.1228$_{\pm 0.0025}$	24.99$_{\pm 0.21}$	0.7948$_{\pm 0.0016}$
✗	✓	0.1141$_{\pm 0.0071}$	25.61$_{\pm 0.66}$	**0.8173**$_{\pm 0.0038}$
✓	✓	**0.1097**$_{\pm 0.0035}$	**26.42**$_{\pm 0.27}$	0.8170$_{\pm 0.0070}$

Table 4. Ablation study of the spatial and temporal positional encodings. The mean and standard deviation are reported for three initializations of the model's parameters. Bold values display the best performance for each metric.

5. Conclusions and Perspectives

In this work, we propose a revised version of GraphCast for forecasting water resources in basins. We use sequences of satellite images with an irregular temporal sampling of approximately two months to predict the next image in the series. A key element is the segmentation map used as a single mesh to train the GNN. In addition to outperforming LSTM, ConvLSTM and TDCNN-ConvLSTM, the model gives good visual results. It also showed stable results for the main hyperparameter, *i.e.*, the number of superpixels.

Although the results are promising, the generalization ability of our revised GraphCast model needs to be assessed on larger datasets. In particular, we would like to further develop the model to large patches, for which multi-scale segmentation, used as a multi-mesh, can help capture long-range dependencies. A further idea is to test the model in the roll-out setting of GraphCast to predict images over an increasing period of time.

For this setting, we believe that longer and denser input time series could help to improve the model's stability.

6. Acknowledgments

The authors thank the French spatial agency (CNES) and the Brittany region (GIS BreTel) for their financial support. This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014108R1 made by GENCI. Charlotte Pelletier is partially funded through project DECOL ANR-23-CE56-0003.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R. et al., 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Cachay, S. R., Erickson, E., Bucker, A. F. C., Pokropek, E., Potosnak, W., Bire, S., Osei, S., Lütjens, B., 2021. The World as a Graph: Improving El Niño Forecasts with Graph Neural Networks. *arXiv preprint arXiv:2104.05089*.

Censi, A. M., Ienco, D., Gbodjo, Y. J. E., Pensa, R. G., Interdonato, R., Gaetano, R., 2021. Attentive spatial temporal graph CNN for land cover mapping from multi temporal remote sensing data. *IEEE Access*, 9, 23070–23082.

Dufourg, C., Pelletier, C., May, S., Lefèvre, S., 2023. Graph Dynamic Earth Net: Spatio-temporal graph benchmark for satellite image time series. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 7164–7167.

Fey, M., Lenssen, J. E., 2019. Fast graph representation learning with PyTorch Geometric. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Guo, X., Lao, J., Dang, B., Zhang, Y., Yu, L., Ru, L., Zhong, L., Huang, Z., Wu, K., Hu, D. et al., 2023. Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery. *arXiv preprint arXiv:2312.10115*.

Interdonato, R., Ienco, D., Gaetano, R., Ose, K., 2019. DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149, 91–104.

Jiao, L., Chen, J., Liu, F., Yang, S., You, C., Liu, X., Li, L., Hou, B., 2022. Graph representation learning meets computer vision: A survey. *IEEE Transactions on Artificial Intelligence*, 4(1), 2–22.

Kavran, D., Mongus, D., Žalik, B., Lukač, N., 2023. Graph Neural Network-Based Method of Spatiotemporal Land Cover Mapping Using Satellite Imagery. *Sensors*, 23(14), 6648.

Keisler, R., 2022. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., Battaglia, P., 2023. Learning skillful medium-range global weather forecasting. *Science*, 382(6677), 1416-1421.

Lin, H., Gao, Z., Xu, Y., Wu, L., Li, L., Li, S. Z., 2022. Conditional local convolution for spatio-temporal meteorological forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 7470–7478.

Mauro, F., Rich, B., Muriga, V. W., Janku, F., Sebastianelli, A., Ullo, S. L., 2023. SEN2DWATER: A novel multispectral and multitemporal dataset and deep learning benchmark for water resources analysis. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 297–300.

McFeeters, S. K., 1996. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing*, 17(7), 1425–1432.

Miller, L., Pelletier, C., Webb, G. I., 2024. Deep Learning for Satellite Image Time Series Analysis: A Review. *arXiv preprint arXiv:2404.03936*.

Moskolaï, W. R., Abdou, W., Dipanda, A., Kolyang, 2021. Application of deep learning architectures for satellite image time series prediction: A review. *Remote Sensing*, 13(23), 4822.

Ning, D., Vetrova, V., Bryan, K. R., Koh, Y. S., 2024. Harnessing the power of graph representation in climate forecasting: Predicting global monthly mean sea surface temperatures and anomalies. *Earth and Space Science*, 11(3), e2023EA003455.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P., 2020. Learning mesh-based simulation with graph networks. *International Conference on Learning Representations*.

Russo, L., Mauro, F., Memar, B., Sebastianelli, A., Gamba, P., Ullo, S. L., 2024. Using Multi-Temporal Sentinel-1 and Sentinel-2 data for water bodies mapping. *arXiv preprint arXiv:2402.00023*.

Rußwurm, M., Körner, M., 2017. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 11–19.

Rußwurm, M., Körner, M., 2018. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS International Journal of Geo-Information*, 7(4), 129.

Rußwurm, M., Pelletier, C., Zollner, M., Lefèvre, S., Körner, M., 2020. Breizhcrops: A time series dataset for crop type mapping. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 43.

Sainte Fare Garnot, V., Landrieu, L., 2021. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4872–4881.

Sun, J., Lai, Z., Di, L., Sun, Z., Tao, J., Shen, Y., 2020. Multilevel deep learning network for county-level corn yield estimation in the us corn belt. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 5048–5060.

Tarasiou, M., Chavez, E., Zafeiriou, S., 2023. ViTs for SITS: Vision transformers for satellite image time series. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10418–10428.

Tseng, G., Zvonkov, I., Purohit, M., Rolnick, D., Kerner, H., 2023. Lightweight, pre-trained transformers for remote sensing timeseries. *arXiv preprint arXiv:2304.14065*.

Tulczyjew, L., Kawulok, M., Longépé, N., Le Saux, B., Nalepa, J., 2022. Graph neural networks extract high-resolution cultivated land maps from Sentinel-2 image series. *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors, 2014. scikit-image: image processing in Python. *PeerJ*, 2, e453. https://doi.org/10.7717/peerj.453.

Voelsen, M., Lauble, S., Rottensteiner, F., Heipke, C., 2023. Transformer Models for Multi-Temporal Land Cover Classification Using Remote Sensing Images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 981–990.

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.

Xu, H., 2006. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *International Journal of Remote Sensing*, 27(14), 3025–3033.