Supplementary Material for: Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks

Vivien Sainte Fare Garnot Loic Landrieu
LASTIG, Univ. Gustave Eiffel, ENSG, IGN, F-94160 Saint-Mande, France
{vivien.sainte-fare-garnot, loic.landrieu}@ign.fr

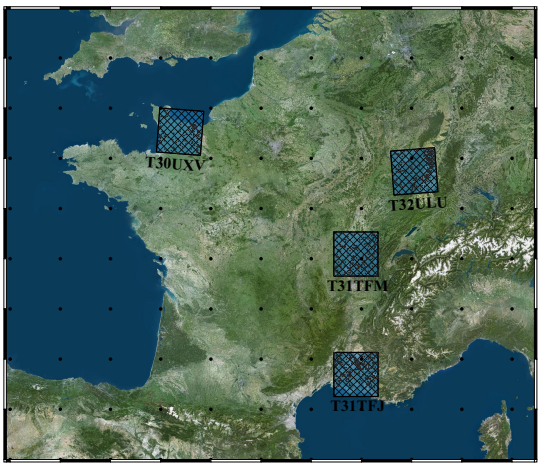
In this appendix, we provide additional information on the PASTIS dataset and our exact model configuration. We also provide complementary qualitative experimental results.

1. PASTIS Dataset

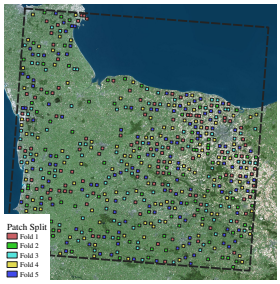
Overview. The PASTIS dataset is composed of 2433 square 128×128 patches with 10 spectral bands and at 10m resolution, obtained from the open-access Sentinel-2 platform. [1] For each patch, we stack all available acquisitions between September 2018 and November 2019, forming our four dimensional multi-spectral SITS: $T \times C \times H \times W$. The publicly available French Land Parcel Identification System (FLPIS) allows us to retrieve the extent and content of all parcels within the tiles, as reported by the farmers. Each patch pixel is annotated with a semantic label corresponding to either the parcels' crop type or the background class. The pixels of each unique parcel in the patch receive a corresponding instance label.

Dataset Extent. The SITS of PASTIS are taken from 4 different Sentinel-2 tiles in different regions of the French metropolitan territory as depicted in Figure 1a. These regions cover a wide variety of climates and culture distributions. Sentinel tiles span 100×100km and have a spatial resolution of 10 meter per pixel. Each pixel is characterized by 13 spectral bands. We select all bands except the atmospheric bands B01, B09, and B10. Each of these tiles is subdivided in square patches of size 1.28×1.28km (128×128 pixels at 10m/pixel), for a total of around $24,000$ patches. We then select $2,433$ patches (10% of all available patches, see Figure 1b), favoring patches with rare crop types in order to decrease the otherwise extreme class imbalance of the dataset.

[1] https://scihub.copernicus.eu



(a) Location of the four tiles.



(b) Selected patches.

(c) Single patch.

Figure 1: **Data Location.** Spatial distribution of the four Sentinel tiles used in PASTIS 1a, and of the selected patches of tile T30UXV 1b. We show an example of patch in 1c, and highlight with red circles examples of parcels that are mostly outside of the patch's extent and thus annotated with the void label. The green circle ○ highlight a parcel partially cut off by the patch borders, but with sufficient overlap to be kept as a valid parcel.

Nomenclature The FLPIS uses a 73 class breakdown for crop types. We select classes with at least 400 parcels and with samples in at least 2 of the 4 Sentinel-2 tiles. This leads us to adopt a 18 classes nomenclature, presented in Figure 2. Parcels belonging to classes not in our 18-classes nomenclature are annotated with the *void* label, see below.

Patch Boundaries. The FLPIS allows us to retrieve the pixel-precise borders of each parcel. We also compute bounding boxes for each parcel. The parcels' extents are cropped along the extent of their 128×128 patch, and the bounding boxes are modified accordingly. Parcels whose surface is more than 50% outside of the patch are annotated with the *void* label, see Figure 1c.

Void and Background Labels. Pixels which are not within the extent of any declared parcel are annotated with the background "stuff" label, corresponding to all non-agricultural land uses. For the semantic segmentation task, this label becomes the 20-th class to predict. In the panoptic setting, this label is associated with pixels not within the extent of any predicted parcel. We do not compute the panoptic metrics for the background class, since our focus is on retrieving the parcels' extent rather than an extensive land-cover prediction. In other words, the reported panoptic metrics are the "things" metrics, which already penalize parcels predicted for background pixels by counting them as false positives.

The void class is reserved for *out-of-scope* parcels, either because their crop type is not in our nomenclature or because their overlap with the selected square patch is too small. We remove these parcels from all semantic or panoptic metrics and losses. Predicted parcels which overlap with an IoU superior to 0.5 with a void parcel are not counted as false positive or true positive, but are simply ignored by the metric, as recommended in [2].

Cross-Validation. The 2,433 selected patches are randomly subdivided into 5 splits, allowing us to perform cross-validation. The official 5-fold cross-validation scheme used for benchmarking is given in Table 1. In order to avoid heterogeneous folds, each fold is constituted of patches taken from all four Sentinel tiles. We also chose folds with comparable class distributions, as measured by their pairwise Kullback-Leiber divergence. We show the resulting class distribution for each fold in Figure 3. Finally, we prevent adjacent patches from being in different folds to avoid data contamination. Geo-referencing metadata of the patches and parcels is included in PASTIS, allowing for the constitution of geographically consistent folds to evaluate spatial generalization. However, this is out of the scope of this paper.

Label and Color	Class Name	Number of parcels
0	*Background*	-
1	Meadow	31292
2	Soft winter wheat	8206
3	Corn	13123
4	Winter barley	2766
5	Winter rapeseed	1769
6	Spring barley	908
7	Sunflower	1355
8	Grapevine	10640
9	Beet	871
10	Winter triticale	1208
11	Winter durum wheat	1704
12	Fruits, vegetables, flowers	2619
13	Potatoes	551
14	Leguminous fodder	3174
15	Soybeans	1212
16	Orchard	2998
17	Mixed cereal	848
18	Sorghum	707
19	*Void label*	35924

Figure 2: Color code of our class nomenclature, and the number of parcel per class.

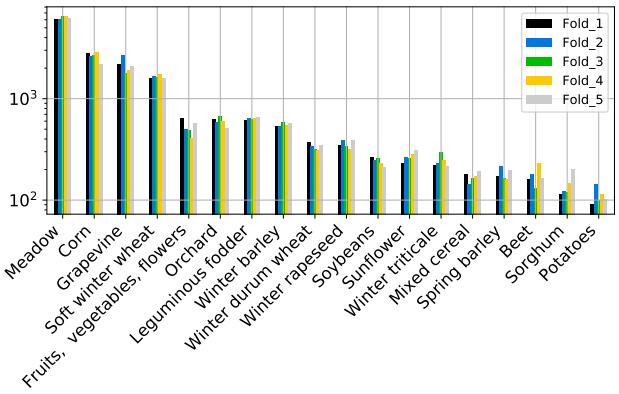


Figure 3: Class distribution for the five folds (in log-scale).

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

Table 1: Official 5-fold cross validation scheme. Each line gives the repartition of the splits into train, validation and test set for each fold.

Temporal Sampling. The temporal sampling of the sequences in PASTIS is irregular: depending on their location, patches are observed a different number of times and at dif-

ferent intervals. This is a result of both the orbit schedule of Sentinel-2 and the policy of Sentinel data providers not to process tile observations identified as covered by clouds for more than 90% of the tile's surface. As this corresponds to the *real world* setting, we decided to leave the SITS as is, and thus to encourage methods that can favourably address this technical challenge. As a result, the proposed SITS are constituted of 33 to 61 acquisitions. In order to assess how our model handles lower sampling frequencies, we limited the number of available acquisitions at inference time[2], and observed a drop of performance of -0.7, -2.0, -5.5, and -14.6 points of mIoU with 32, 24, 16, and 8 available dates, respectively.

Clouds Cover. Even after the automatic filtering of predominantly cloudy acquisitions, some patches are still partially or completely obstructed by cloud cover. We opt to not apply further pre-processing or cloud detection, and produce the raw data in PASTIS. Our reasoning is that an adequate algorithm should be able to learn to deal with such acquisitions. Indeed, robustness to cloud-cover has been experimentally demonstrated for deep learning methods by Rußwurm and Körner [3, 4].

2. Implementation Details

In this section, we detail the exact configuration of our method as well as the competing algorithms evaluated.

Training Details. Across our experiments, we use Adam [1] optimizer with default parameters and a batch size of 4 sequences. The semantic segmentation experiments use a fixed learning rate of 0.001 for 100 epochs. For the panoptic segmentation experiments, we start with a higher learning rate of 0.01 for 50 epochs, and decrease it to 0.001 for the last 50 epochs.

U-TAE. In Table 2, we report the width of the feature maps outputted by each level of the U-TAE's encoder and decoder. In both networks, we use the the same convolutional block shown in Figure 4 and constituted of one 3×3 convolution from the input to the output's width, and one residual 3×3 convolution. In the encoding branch, we use Group Normalisation with 4 groups and Batch Normalisation in the decoding branch .

For the temporal encoding, we chose a L-TAE with 16 heads, and a key-query space of dimension $d_k = 4$. We use Group Normalisation with 16 groups at the input and output of the L-TAE, meaning that that the inputs of each head are layer-normalized.

[2]This can be interpreted as the test set having an increased cloud cover.

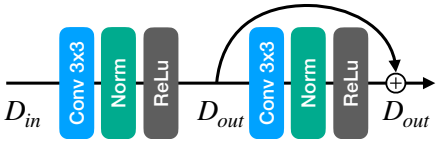


Figure 4: Structure of the convolutional block used in the spatial encoder-decoder network. This block maps a feature map with D_{in} channels to a feature map with D_{out} channels.

Table 2: Width of the feature maps outputted at each level of the encoding and decoding branches of the spatial module.

Encoder		Decoder	
e_1	64	d_1	32
e_2	64	d_2	32
e_3	64	d_3	64
e_4	128	d_4	128

Recurrent Models. We use the same U-Net architecture for our models and *U-BiConvLSTM* and *U-ConvLSTM*, but simply replace the L-TAE by a ConvLSTM or BiConvLSTM respectively. The hidden state's size of the biConvLSTM is chosen as 32 in both directions, and 64 for the convLSTM. For the recurrent-convolutional methods *ConvLSTM* and *ConvGRU* not using a U-Net, we set hidden sizes of 160 and 188 respectively.

3D-Unet. For this network, we use the official PyTorch implementation of Rustowicz *et al.* [5]. This network is constituted of five successive 3D-convolution blocks with spatial down-sampling after the 2nd and 4th blocks. Each convolutional block doubles the number of channels of the processed feature maps, and the innermost feature maps have a channel dimension set to 128. Leaky ReLu and 3D Batch Normalisation are used across the convolutional blocks of this architecture. The sequence of feature maps is averaged along the temporal dimension to produce the final embedding of the input image sequence. In their implementation, the authors used a linear layer to collapse the temporal dimension, yet this was not a valid option for PASTIS as the sequences have highly variable lengths and the sequence indices do not correspond to the same acquisition date from one sequence to another.

FPN-ConvLSTM. For this architecture, the input sequence of images is first mapped to feature maps of channel dimension 64 with two consecutive 3×3 convolution layers, followed by Group Normalization and ReLu. A 5-level feature pyramid is then constructed for each date of the sequence by applying to the feature maps 4 different 3×3 convolution of respective dilation rates 1, 2, 4 and 8, and

Table 3: Configuration of the four MLPs of PaPs

MLP	Layers	Final Layer
Shape	$256 \mapsto 128 \mapsto S^2$	-
Size	$256 \mapsto 128 \mapsto 2$	Softplus
Class	$256 \mapsto 128 \mapsto 64 \mapsto K$	Softmax

computing the spatial average of the feature map. These 5 maps are concatenated along the channel dimension, and processed by a ConvLSTM with a hidden state size of 88. We found it beneficial to use a supplementary convolution before the ConvLSTM to reduce the number of channels of the feature pyramid by a factor 2.

PaPs module. In the PaPs module, the saliency and heatmap predictions are obtained with two separate convolutional blocks operating on the high resolution feature map d_1 with 32 channels. These blocks are composed of two convolutional layers of width 32 and 1 respectively. We use Batch Normalisation and ReLu after the first convolution, and a sigmoid after the second.

The 256-dimensional multi-scale feature vector $(128 + 64 + 32 + 32)$ is mapped to the shape, class and size predictions by three different MLPs described in Table 3. The inner layers use Batch Normalisation and ReLu activation.

The residual CNN used for shape refinement is composed of three convolutional layers : $1 \mapsto 16 \mapsto 16 \mapsto 1$, with ReLu activation and instance normalisation on the first layer only.

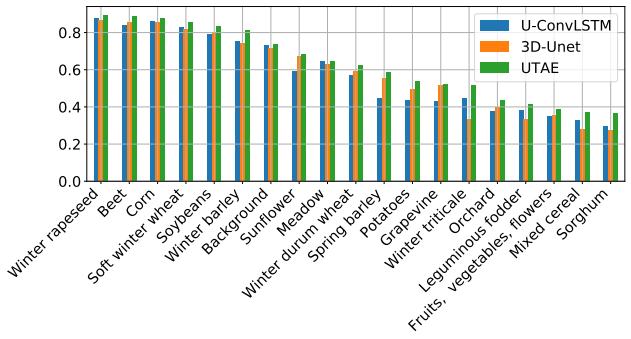


Figure 5: Per class IoU of the three best performing semantic segmentation models. Our U-TAE outperforms the other two approaches on every classes, and brings noticeable improvement on hard classes such as Mixed cereal and Sorghum.

Handling Sequences of Variable Lengths. All models are trained on batches of sequences of variable length. To facilitate the handling of batches by the GPU, we append all-zeroes images at the end of shorter sequences to match the length of the longer sequence in the batch. We retain a padding mask to prevent the spatial and temporal encoding of padded values, and to exclude these padded values from temporal averages.

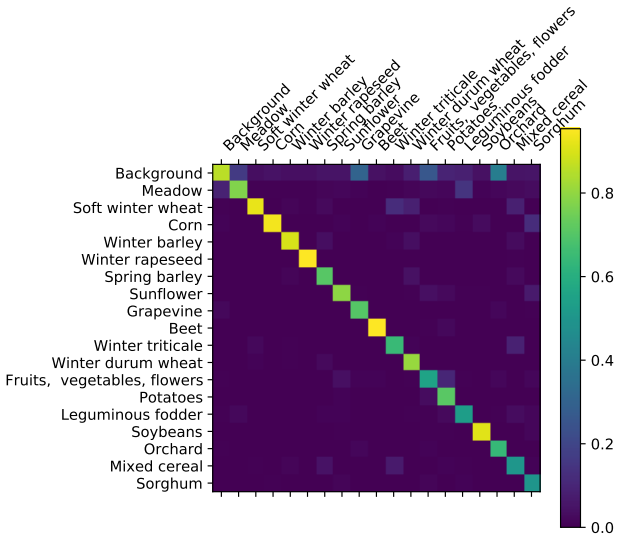


Figure 6: Confusion matrix of U-TAE for semantic segmentation on PASTIS. The color of each pixel at line i and column j corresponds to the proportion of samples of the class i that were attributed to the class j.

3. Additional Results

In Figure 5, we show the class-wise performance of the three best performing semantic segmentation models, displaying an improvement of U-TAE compared to the other methods across all crop types. We also show on Figure 6 the confusion matrix of U-TAE. Unsurprisingly, confusions seem to occur between semantically close classes such as different cereal types, or *Sunflower* and *Fruits, Vegetable, Flower*.

In Figure 7, we present qualitative results illustrating the predicted panoptic and semantic segmentations compared to the ground truth. In particular, we show some failure cases in which thin or visually fragmented parcels are not recovered correctly.

In Figure 8, we illustrate the results of the semantic segmentation for our method and three other competing approaches: *3D-Unet*, *U-BiConvLSTM*, and *convGRU*. We show how our multi-scale temporal attention masks allow our predictions to be both pixel-precise and consistent for large parcels.

Finally, we present in Figure 9 an example of inference using a single image from the sequence. As expected for mono-temporal segmentation, the parcel classification is poor. Furthermore, we show a case of a border that is essen-

tially invisible on a single image, but that our full model is able to detect using the entire sequence of satellite images.

Acknowledgments

The satellite images used in PASTIS were gathered from THEIA: "*Value-added data processed by the CNES for the Theia data cluster using Copernicus data. The treatments use algorithms developed by Theia's Scientific Expertise Centres.*" The annotations of PASTIS were taken from the French LPIS produced by IGN, the French mapping agency. This work was partly supported by ASP, the French Payment Agency.

References

[1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[2] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019.

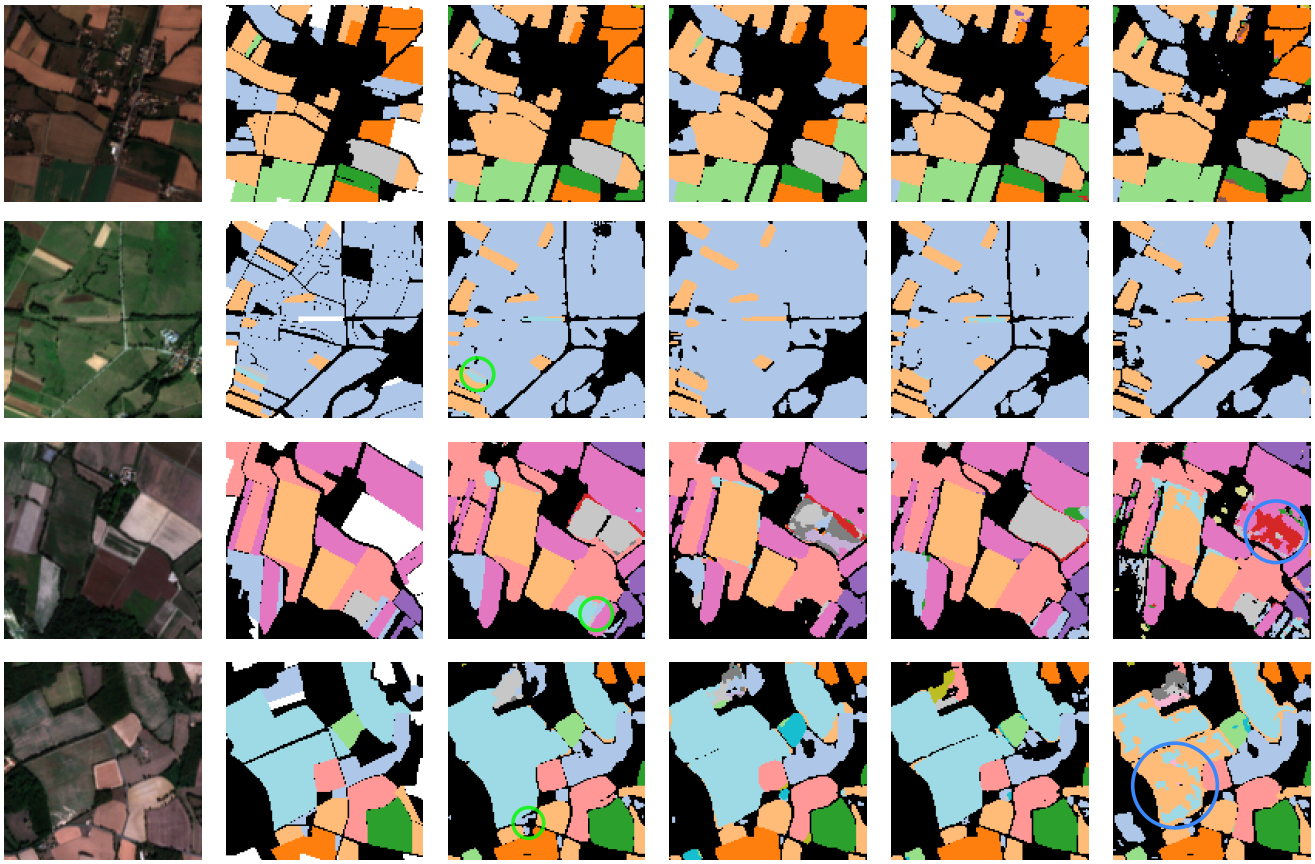
[3] Marc Rußwurm and Marco Körner. Convolutional LSTMs for cloud-robust segmentation of remote sensing imagery. *NeurIPS Workshops*, 2018.

[4] Marc Rußwurm and Marco Körner. Self-attention for raw optical satellite time series classification. *ISPRS*, 2020.

[5] Rose Rustowicz, Robin Cheong, Lijing Wang, Stefano Ermon, Marshall Burke, and David Lobell. Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. In *CVPR Workshops*, 2019.

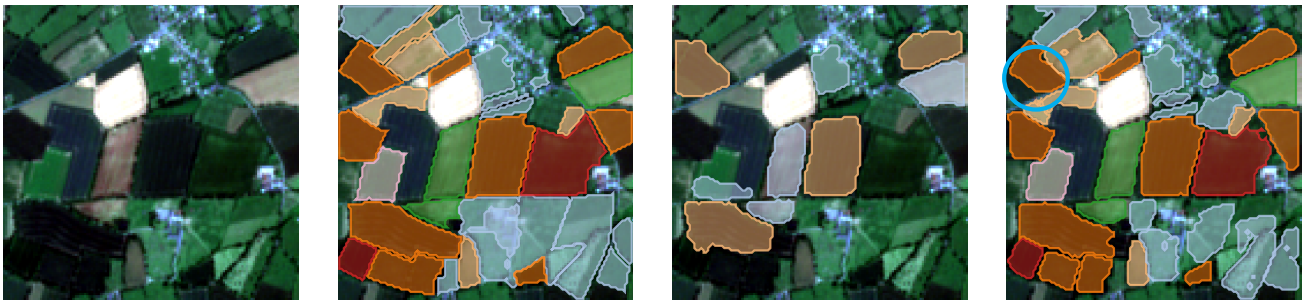
(a) Image from the sequence. (b) Panoptic annotation. (c) Panoptic segmentation. (d) Semantic segmentation.

Figure 7: **Qualitative Panoptic Segmentation Results.** We represent a single image from the sequence using the RGB channels (a), and whose ground truth parcel's limit and types are known (b). We then represent the parcels predicted by our panoptic segmentation module (c), and the pixelwise prediction of our semantic segmentation module (d). See Figure 2 for the color to crop type correspondence. We highlight with a green circle ○ a large, fragmented parcel declared as one single field. This leads to predictions with low confidence and a low panoptic quality. Conversely, the cyan circle ○ highlights such fragmented parcel which is correctly predicted as a single instance. This suggests that our network is able to use the temporal dynamics to recover ambiguous borders. We highlight a failure case with the red circle ○, for which many thin parcels are not properly detected, resulting in a low panoptic quality. We observe that the semantic segmentation model struggles as well for such thin parcels. Finally, we highlight with a blue circle ○ an example in which the panoptic prediction is superior to the semantic segmentation, indicating that detecting parcels' boundaries and extent can be informative for their classification.



(a) Single image. (b) Annotation. (c) U-TAE. (d) 3D-Unet. (e) U-BiConvLSTM. (f) ConvGRU.

Figure 8: **Qualitative Semantic Segmentation Results.** We represent a single image from the sequence using the RGB channels (a), and whose ground truth parcel's limit and crop type are known (b). We then represent the pixelwise prediction from our approach (c), and for three other competing algorithms (d-f). The different predictions shown on this figure illustrate the importance of the resolution at which temporal encoding is performed. ConvGRU applies a recurrent-convolutional network at the highest resolution, which results in predictions with high spatial variability. As a consequence, the prediction over large parcels are inconsistent (blue circles ○). Conversely, U-BiConvLSTM applies temporal encoding to feature maps with a larger receptive field, resulting in more spatially consistent predictions. Yet, this architecture often fails to retrieve small or thin parcels. In contrast, our U-TAE produces spatially consistent predictions on large parcels, while being able to retrieve such small parcels (green circles ○). 3D-Unet also uses temporal encoding at different resolution levels, yet fails to recover these small parcels.



(a) Single observation. (b) Panoptic annotation. (c) Mono-temporal prediction. (d) Multi-temporal prediction.

Figure 9: **Mono-temporal Panoptic Segmentation.** We train our mono-temporal model on a single image (a), with panoptic annotation (b). We then compare the results of the mono-temporal model in (c) with the results our full model when performing inference on the full length sequence (d) from which the single patch (a) is drawn. First, we observe that many parcels are not detected by the mono-temporal model, indicating an overall low predicted quality. Second, we can see that most detected parcels are misclassified by the mono-temporal model. This is in accordance with the low semantic segmentation score of the mono-temporal model: crop types are hard to distinguish from a single observation. Last, adjacent parcels with no clear borders are predicted as a single parcel, when the multi-temporal model is able to differentiate between the two parcels (cyan circle ○). This illustrates how using SITS instead of single images can help resolve ambiguous parcels delineation.