Lightweight, Pre-trained Transformers for Remote Sensing Timeseries

Gabriel Tseng[1,2] **Ruben Cartuyvels**[1,3] **Ivan Zvonkov**[4] **Mirali Purohit**[5]
David Rolnick[1,2] **Hannah Kerner**[5]
[1] Mila – Quebec AI Institute
[2] McGill University
[3] KU Leuven
[4] University of Maryland, College Park
[5] Arizona State University

Abstract

Machine learning models for parsing remote sensing data have a wide range of societally relevant applications, but labels used to train these models can be difficult or impossible to acquire. This challenge has spurred research into self-supervised learning for remote sensing data. Current self-supervised learning approaches for remote sensing data draw significant inspiration from techniques applied to natural images. However, remote sensing data has important differences from natural images – for example, the temporal dimension is critical for many tasks and data is collected from many complementary sensors. We show we can create significantly smaller performant models by designing architectures and self-supervised training techniques specifically for remote sensing data. We introduce the **P**retrained **Re**mote **S**ensing **T**rans**fo**rmer (Presto), a transformer-based model pre-trained on remote sensing pixel-timeseries data. Presto excels at a wide variety of globally distributed remote sensing tasks and performs competitively with much larger models while requiring far less compute. Presto can be used for transfer learning or as a feature extractor for simple models, enabling efficient deployment at scale.

1 Introduction & Related Work

Recent advances in machine learning capabilities combined with vast remote sensing datasets have provided critical tools for mitigating and adapting to climate change, ranging from improved weather forecasting (English et al., 2013; Voosen, 2020) to disaster management (Kansakar and Hossain, 2016) to improving food security in a changing climate Krafft; Tseng et al. (2020). However, the remote sensing data modality has several characteristics that are important to consider when designing machine learning algorithms in this domain:

- **Highly multi-modal data**: Satellites carry a wide range of sensors, including synthetic aperture radar (Torres et al., 2012) and multispectral optical sensors (Drusch et al., 2012). In addition, there are many derived data products which are created by the manipulation of these raw data sources (such as digital elevation maps (Rabus et al., 2003)).
- **A highly informative temporal dimension**: The Earth's highly dynamic nature (Yifang et al., 2015) and the relatively coarse resolution of freely available satellite data means that in remote sensing, the temporal dimension is critical for many downstream tasks (Rußwurm et al., 2023). A common approach by remote sensing practitioners is therefore to train single pixel-timeseries models (Rußwurm et al., 2023; Sainte Fare Garnot et al., 2020; Pelletier et al., 2019; Wang et al., 2020; Hengl et al., 2017)

Tackling Climate Change with Machine Learning Workshop at the 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

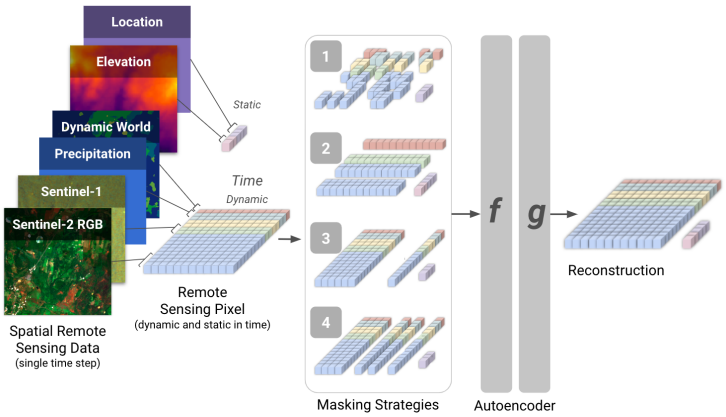


Figure 1: Presto learns from structurally-masked remote sensing pixel-timeseries.

Datasets for remote sensing often have very few labels (Helber et al., 2019), which may be unreliable (Bressan et al., 2022; Yifang et al., 2015) or absent, especially for under-resourced geographies (Kerner et al., 2020; Nakalembe et al., 2021), leading to poor global generalization (Yifang et al., 2015). This limited availability of labels combined with plentiful unlabeled data has spurred the investigation of self-supervised learning algorithms for remote sensing data (Reed et al., 2022; Cong et al., 2022; Jean et al., 2019; Manas et al., 2021; Ayush et al., 2021). Previous approaches investigating self-supervised learning for remote sensing primarily treat remote sensing data as analogous to natural imagery, and therefore attempt to co-opt methods and architectures originally designed for natural imagery (i.e., ground-level photography) – for example, by using a ResNet (He et al., 2016) backbone (Jean et al., 2019; Manas et al., 2021; Ayush et al., 2021), or by adapting masked autoencoding for image classification (He et al., 2022) to satellite imagery (Reed et al., 2022; Cong et al., 2022). These models fail to leverage all the attributes of remote sensing data (for example, models which can only ingest data from a single RGB sensor, or which do not consider the temporal dimension of the data). While several pre-training methods have been proposed for remote sensing timeseries (Yuan and Lin, 2020; Yuan et al., 2022, 2023), these have not aimed at multi-task, global applicability, having been pre-trained and evaluated on highly local areas (e.g., central California) and evaluated only for a single task (e.g., crop type classification).

To take advantage of the unique characteristics, global scope, and broad applicability of remote sensing data, we introduce the **P**retrained **Re**mote **S**ensing **T**ransf**o**rmer (Presto), a lightweight transformer-based model designed to ingest pixel-timeseries inputs from a variety of Earth observation sensors and data products. We tailor the self-supervised learning process to learn from multiple data sources and from the temporal dimension of the data so that Presto learns powerful representations of remote sensing data. These representations can be efficiently adapted to a wide range of globally distributed remote sensing tasks. Presto is also robust to missing input sensors or timesteps, excelling even in image-based tasks where the temporal dimension is completely absent.

Models for remote sensing data are typically used to make contiguous geospatial predictions over millions (or billions) of samples to form a predicted map. The computational performance of models is therefore one of the primary considerations at deployment time (Van Tricht, 2021; Hengl et al., 2017; Robinson et al., 2019). This has limited the adoption of large models with ViT or ResNet backbones for large scale mapping. In comparison, Presto is highly accurate despite having $1000\times$ fewer parameters than ViT- or ResNet-based models, making it well-suited to real-world deployment.

2 Method

We aim to learn a model, f, which can learn useful representations in a self-supervised manner given unlabelled remote sensing pixel-timeseries data. Our approach is based on the masked autoencoding framework (He et al., 2022), in which the network architecture includes both an encoder (f) and a decoder (g). During pre-training, part of the input is masked out and the encoder embeds the remaining (non-masked) part of the input. The decoder uses this to reconstruct the masked-out part of

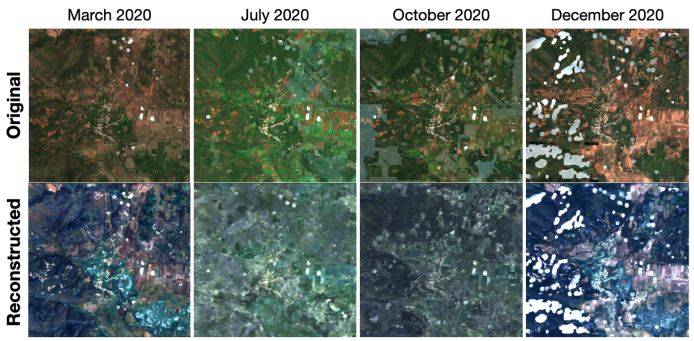


Figure 2: **Presto learns to reconstruct channels that are completely masked in a spatially cohesive manner**. We mask the Sentinel-2 RGB channels; Presto is able to reconstruct these channels when they are completely absent from the input. We note that the outputs are spatially consistent, even though Presto only processes pixel-timeseries.

the input. For downstream tasks, we discard g and only use f as a feature extractor or a fine-tuneable model. The pre-training set up is shown in Figure 2, with additional details in Appendix A.1.

Pre-training Data

We collect a dataset to pre-train Presto that is highly diverse in terms of sensor types and in terms of geographic and semantic diversity, so that Presto can be applied to many different downstream tasks. We follow the sampling strategy of Dynamic World (Brown et al., 2022) to obtain globally representative samples, yielding 21.5M pixel samples at 10 m/pixel resolution. For each sample, we take a 2-year monthly pixel-timeseries from the beginning of 2020 to the end of 2021. In addition to global coverage, we also use a range of directly-sensed (e.g. Sentinel-1 synthetic aperture radar imagery Torres et al. (2012)) and derived (e.g. Dynamic World landcover classes Brown et al. (2022)) Earth observation products, exported using Google Earth Engine (Gorelick et al., 2017). Appendix A.1.1 describes the data in detail.

Encoding and Tokenization

An input pixel-timeseries is transformed into a number of tokens to be processed by the Presto transformer. The input variables are split into channel groups according to their source: e.g., the S1 bands form one channel group. Each channel group is projected to the token-space by their own learnt linear projection h: e.g., $e_i^{\text{S1}} = h^{\text{S1}}(t_i^{\text{S1}})$. We add encodings to the tokens to communicate a token's (i) timestamp and (ii) channel group. The complete encoding is a concatenation of the following positional, month and learnt channel encodings:

- **Positional**: The sinusoidal encoding p_{sin} originally introduced by Vaswani et al. (2017).
- **Month**: Another sinusoidal encoding p_{month} that represents the month being captured by each token (vs. the position in the input timeseries), because we expect timesteps from similar months to be similar even if they are from different years.
- **Channel Group**: Finally, each token is associated with a set of input channels. We apply a learnable encoding p_{channel} for each channel group.

Pre-training via Structured Masking

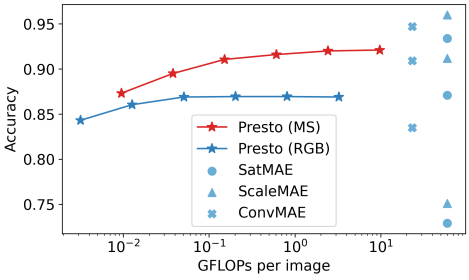
A goal of this model is to perform well even with incomplete inputs. We therefore tailor the masking strategies to encourage the model to learn representations that perform well when given only a subset of bands or timesteps for downstream tasks. For each $T \times D$ input sample of T timesteps and D total input channels, we randomly sample one of the following masking techniques: (i) **Random**: $(t \times d)$ masked values, with $t < T$ and $d < D$, (ii) **Channel-groups**: $(T \times d)$ masked values, with $d < D$, (iii) **Contiguous timesteps**: $(t \times D)$ masked values, with $t < T$, (iv) **Random timesteps**: $(t \times D)$ masked values, with $t < T$.

Table 1: **Adapting Presto to downstream tasks is computationally efficient**. We show F1 scores on the CropHarvest tasks. TIML and MOSAIKS-1D do not receive Dynamic World as input, so we evaluated Presto with and without it for fair comparison. In both cases, Presto outperforms these models while requiring the adaptation of far fewer parameters.

Model	Number of parameters		F1 Score			
	Total	Finetuned	Kenya	Brazil	Togo	Mean
Random Forest			0.559	0.000	0.756	0.441
MOSAIKS-1D$_R$	418K	8193	0.790	0.746	0.679	0.738
TIML	91K	91K	0.838	0.835	0.732	0.802
Presto$_R$	401K	129	0.816	**0.891**	**0.798**	0.835
no DW			**0.861**	0.888	0.760	**0.836**

	16	32	64	Average
SatMAE	0.729	0.871	0.934	0.845
ScaleMAE	0.751	0.912	**0.960**	0.867
ConvMAE	0.835	0.909	0.947	0.888
Presto (RGB)	0.869	0.869	0.869	0.869
Presto (MS)	**0.916**	**0.920**	0.921	**0.919**

(a)



(b)

Table 2: **Presto is competitive with MAE methods designed for single-timestep satellite images, while being much more computationally efficient**. We plot (a) EuroSAT accuracy as a function of input image resolution, and (b) as a function of FLOPs required to encode an image (note the log scale on the x-axis). The image-based models resize all images to 224×224, so the FLOPs required to encode an image do not change as image resolution changes. As in (Reed et al., 2022), we compute these results by running a KNN@5 classifier on the output encodings of the models. In Figure (b), we additionally run Presto at resolutions $\{2, 4, 8\}$ - full results are available in the appendix. Presto achieves competitive results with other MAE models while requiring **up to three orders of magnitude less FLOPs to encode an image**.

3 Results & Discussion

We test Presto on a wide variety of downstream tasks (full results are available in the appendix). We focus here on two downstream tasks which demonstrate Presto's performance and flexibility:

- **Crop type Segmentation**: The CropHarvest (Tseng et al., 2021b) evaluation datasets consist of classifying **pixel-timeseries** as (i) maize in Kenya, (ii) coffee in Brazil and (iii) crop or non-crop in Togo. We compare Presto to the baselines which accompany CropHarvest and to Task-Informed Meta-Learning (TIML, Tseng et al., 2021a), a meta-learning method which achieves state-of-the-art results on these datasets. We train a logistic regression on the frozen Presto model's outputs. Presto outperforms TIML on this dataset (Table 1), while requiring the adaptation of far fewer parameters.
- **EuroSAT**: The EuroSAT dataset consists of classifying **single-timestep images** in Europe as belonging to one of 10 landcover classes (Helber et al., 2019). We use the train and test splits provided by Neumann et al. (2019). We compare Presto to SatMAE, ConvMAE and ScaleMAE by using the KNN-classifier approach at a variety of input resolutions, as is done by ScaleMAE (Reed et al., 2022). Since Presto is designed to ingest pixel-timeseries (and not single timestep images), we pass the mean and standard deviation of Presto's outputs per-pixel to the KNN-classifier. Despite the difference of image inputs from Presto's pretraining data, Presto is competitive with these much larger models, while requiring orders of magnitude less compute to encode images (Table 2).

Conclusion We present Presto: a lightweight, pre-trained timeseries transformer for remote sensing. By leveraging structure unique to remote sensing data we are able to train an extremely lightweight

model which achieves state-of-the-art results in a wide variety of globally distributed evaluation tasks. Computational efficiency is of paramount importance in remote sensing settings (often dictating which models ultimately get selected for deployment). We demonstrate that strong performance can be achieved while meeting this constraint, and that self-supervised learning can provide significant benefits even for small models.

References

Tick tick bloom: Harmful algal bloom detection challenge. https://www.drivendata.org/competitions/143/tick-tick-bloom/page/649/, 2023. Accessed: 2023-03-10.

S. Ahlswede, C. Schulz, C. Gava, P. Helber, B. Bischke, M. Förster, F. Arias, J. Hees, B. Demir, and B. Kleinschmit. Treesatai benchmark archive: A multi-sensor, multi-label dataset for tree species classification in remote sensing. *Earth System Science Data*, 2023.

K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon. Geography-aware self-supervised learning. In *CVPR*, 2021.

V. Böhm, W. J. Leong, R. B. Mahesh, I. Prapas, E. Nemni, F. Kalaitzis, S. Ganju, and R. Ramos-Pollan. Sar-based landslide classification pretraining leads to better segmentation. In *Artificial Intelligence for Humanitarian Assistance and Disaster Response Workshop at NeurIPS*, 2022.

P. O. Bressan, J. M. Junior, J. A. C. Martins, M. J. de Melo, D. N. Gonçalves, D. M. Freitas, A. P. M. Ramos, M. T. G. Furuya, L. P. Osco, J. de Andrade Silva, et al. Semantic segmentation with labeling uncertainty and class imbalance applied to vegetation mapping. *International Journal of Applied Earth Observation and Geoinformation*, 2022.

C. F. Brown, S. P. Brumby, B. Guzder-Williams, T. Birch, S. B. Hyde, J. Mazzariello, W. Czerwinski, V. J. Pasquarella, R. Haertel, S. Ilyushchenko, K. Schwehr, M. Weisse, F. Stolle, C. Hanson, O. Guinan, R. Moore, and A. M. Tait. Dynamic world, near real-time global 10 m land use land cover mapping. *Scientific Data*, Jun 2022.

Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. B. Lobell, and S. Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *NeurIPS*, 2022. URL `https://openreview.net/forum?id=WBhqzpF6KYH`.

M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 2012.

S. English, T. McNally, N. Bormann, K. Salonen, M. Matricardi, A. Moranyi, M. Rennie, M. Janisková, S. Di Michele, A. Geer, et al. Impact of satellite data, 2013.

N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

T. Hengl, J. Mendes de Jesus, G. B. Heuvelink, M. Ruiperez Gonzalez, M. Kilibarda, A. Blagotić, W. Shangguan, M. N. Wright, X. Geng, B. Bauer-Marschallinger, et al. Soilgrids250m: Global gridded soil information based on machine learning. *PLoS one*, 2017.

N. Jean, S. Wang, A. Samar, G. Azzari, D. Lobell, and S. Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *AAAI*, 2019.

P. Kansakar and F. Hossain. A review of applications of satellite earth observation data for global societal benefit and stewardship of planet earth. *Space Policy*, 2016.

H. Kerner, G. Tseng, I. Becker-Reshef, C. Nakalembe, B. Barker, B. Munshell, M. Paliyam, and M. Hosseini. Rapid response crop maps in data sparse regions. In *ACM SIGKDD Conference on Data Mining and Knowledge Discovery Workshops*, 2020.

A. Krafft. ASU researcher combats food insecurity with AI. `https://news.asu.edu/20230303-solutions-asu-researcher-combats-food-insecurity-ai`. Accessed: 2023-09-21.

O. Manas, A. Lacoste, X. Giró-i Nieto, D. Vazquez, and P. Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *CVPR*, 2021.

C. Nakalembe, C. Justice, H. Kerner, C. Justice, and I. Becker-Reshef. Sowing seeds of food security in africa. *Eos (Washington. DC)*, 102, 2021.

M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby. In-domain representation learning for remote sensing. *arXiv preprint arXiv:1911.06721*, 2019.

C. Pelletier, G. I. Webb, and F. Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 2019.

B. Rabus, M. Eineder, A. Roth, and R. Bamler. The shuttle radar topography mission—a new class of digital elevation models acquired by spaceborne radar. *ISPRS journal of photogrammetry and remote sensing*, 2003.

K. Rao, A. P. Williams, J. F. Flefil, and A. G. Konings. Sar-enhanced mapping of live fuel moisture content. *Remote Sensing of Environment*, 2020.

C. J. Reed, R. Gupta, S. Li, S. Brockman, C. Funk, B. Clipp, S. Candido, M. Uyttendaele, and T. Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. *arXiv preprint arXiv:2212.14532*, 2022.

C. Robinson, L. Hou, K. Malkin, R. Soobitsky, J. Czawlytko, B. Dilkina, and N. Jojic. Large scale high-resolution land cover mapping with multi-resolution data. In *CVPR*, 2019.

J. W. Rouse, R. H. Haas, J. A. Schell, D. W. Deering, et al. Monitoring vegetation systems in the great plains with erts. *NASA Spec. Publ*, 351(1):309, 1974.

M. Rußwurm, N. Courty, R. Emonet, S. Lefèvre, D. Tuia, and R. Tavenard. End-to-end learned early classification of time series for in-season crop type mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2023. URL `https://www.sciencedirect.com/science/article/pii/S092427162200332X`.

V. Sainte Fare Garnot, L. Landrieu, S. Giordano, and N. Chehata. Satellite image time series classification with pixel-set encoders and temporal self-attention. *CVPR*, 2020.

R. Torres, P. Snoeij, D. Geudtner, D. Bibby, M. Davidson, E. Attema, P. Potin, B. Rommen, N. Floury, M. Brown, et al. Gmes sentinel-1 mission. *Remote sensing of environment*, 2012.

G. Tseng, H. Kerner, C. Nakalembe, and I. Becker-Reshef. Annual and in-season mapping of cropland at field scale with sparse labels. In *Tackling Climate Change with Machine Learning workshop at NeurIPS*, 2020.

G. Tseng, H. Kerner, and D. Rolnick. TIML: Task-informed meta-learning for crop type mapping. In *AI for Agriculture and Food Systems at AAAI*, 2021a.

G. Tseng, I. Zvonkov, C. L. Nakalembe, and H. Kerner. Cropharvest: A global dataset for crop-type classification. In *NeurIPS, Datasets and Benchmarks Track*, 2021b. URL `https://openreview.net/forum?id=JtjzUXPEaCu`.

K. Van Tricht. Mapping crops at global scale! what works and what doesn't? `https://blog.vito.be/remotesensing/worldcereal-benchmarking`, 2021. Accessed: 2023-07-31.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 2017.

P. Voosen. Europe builds 'digital twin'of earth to hone climate forecasts, 2020.

S. Wang, S. Di Tommaso, J. M. Deines, and D. B. Lobell. Mapping twenty years of corn and soybean across the us midwest using the landsat archive. *Scientific Data*, 2020.

B. Yifang, P. Gong, and C. Gini. Global land cover mapping using earth observation satellite data: Recent progresses and challenges. *ISPRS journal of photogrammetry and remote sensing*, 2015.

Y. Yuan and L. Lin. Self-supervised pretraining of transformers for satellite image time series classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:474–487, 2020.

Y. Yuan, L. Lin, Q. Liu, R. Hang, and Z.-G. Zhou. Sits-former: A pre-trained spatio-spectral-temporal representation model for sentinel-2 time series classification. *International Journal of Applied Earth Observation and Geoinformation*, 106:102651, 2022.

Y. Yuan, L. Lin, Z.-G. Zhou, H. Jiang, and Q. Liu. Bridging optical and sar satellite image time series via contrastive feature extraction for crop classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 195:222–232, 2023.

Channel Groups	Random	Random Timesteps	Contiguous Timesteps	F1 Score
✓				0.646
	✓			0.653
		✓		0.664
			✓	0.649
✓	✓	✓	✓	**0.665**

Table 3: **Structured masking strategies yield the best downstream performance**. We measured Presto$_R$'s F1 score on the CropHarvest validation task. Combining structured strategies outperformed the "Random" masking employed by He et al. (2022).

A Appendix

A.1 Pre-training details

We outline training hyperparameters below:

- **Training length**: We train the model for 20 epochs, with a batch size of 4096 (resulting in 5950 batches per epoch). On a single NVIDIA V100 GPU, this takes 43 $\frac{1}{4}$ hours.
- **Optimizer and learning rate**: We train the model with an AdamW optimizer. We use a cosine annealing schedule for our learning rate, with a maximum learning rate of 0.001 at the 2$^{\text{nd}}$ epoch. We apply a weight decay of 0.05, and βs of (0.9, 0.95).
- **Masking**: We use a masking ratio of 0.75, randomly selecting (for each instance) a masking strategy from the ones described in Section 2. If the masking strategy cannot mask the right number of tokens, we randomly mask additional tokens to achieve the correct masking ratio.

In addition, we include a diagram of the training process in Figure 1. We show results of an ablation study demonstrating the usefulness of the structured masking process in Table 3.

A.1.1 Pretraining data

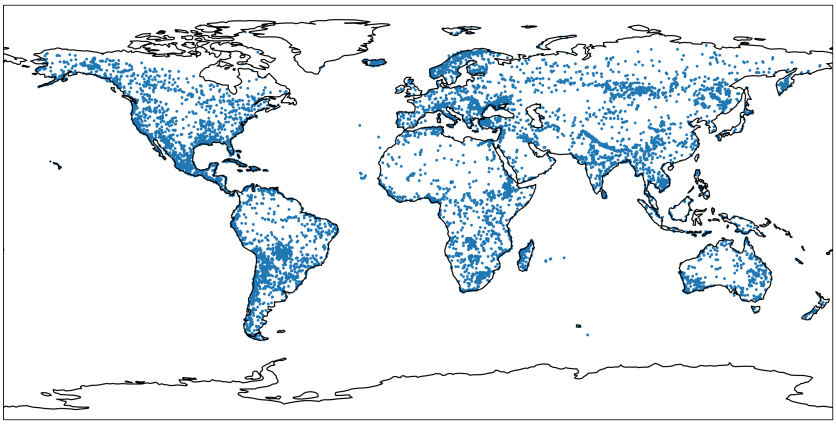


Figure 3: The distribution of the pre-training dataset described in Section 2.

Remote sensing models can be deployed in a wide range of geographies, with few labelled datapoints available at fine-tuning time (Kerner et al., 2020; Böhm et al., 2022). We therefore aim to collect a globally representative pre-training dataset. We achieve this by following the sampling strategy used by Dynamic World (Brown et al., 2022). We divide the Earth into three regions: the Western Hemisphere and two regions in the Eastern Hemisphere. These regions are further divided into ecoregions, and stratified samples are gathered from each region using land cover classes as sampling strata. Figure 3 shows the resulting geographical distribution. Each sample represents a 510×510 pixel tile with a spatial resolution of 10 meter per pixel. To obtain pixel-timeseries we grid-sample 2,500 pixels from each sample, yielding a total of 21,535,000 pixel samples (each with 24 one-month timesteps).

A pre-training batch contains a number of pixel-timeseries, each of which consists of a concatenation of dynamic-in-time datapoints with each timestep representing a month (yielding $T = 12$ timesteps

Table 4: FLOPs required to encode a single EuroSat image (or pixel, for Presto), as measured by the `thop` library. When plotting results in Table 2, we multiply the FLOPs for Presto by the number of pixels encoded for an image. At its highest resolution, EuroSAT images are 64×64, so Presto FLOPs for a full resolution image can be obtained by multiplying the per-pixel FLOPs by 4,096. We include this value in brackets for completeness.

				Presto	
	SatMAE	ScaleMAE	ConvMAE	RGB pixel (image)	MS pixel (image)
MegaFLOPs	59,685.69	59,685.69	23,315.58	0.79 (3,235.84)	2.37 (9,707.52)

in total). To every pixel-timeseries we append a number of static-in-time features. We leverage the following data products when pretraining Presto:

- **Sentinel-1 Synthetic Aperture Radar observations** (S1): The VV (emit and receive at vertical polarization) and VH (emit at vertical and receive at horizontal polarization) bands: 2 real-valued dynamic values per monthly timestep.
- **Sentinel-2 Multispectral images** (S2): We removed the 60m resolution bands, yielding bands with 10m and 20m resolution with channels in the visible, near-infrared and short-wave infrared range: 10 real-valued dynamic values per timestep.
- **ERA5 Climate Reanalysis Meteorological data** (ERA5): Monthly total precipitation and temperature at 2 metres above the ground: 2 real-valued dynamic values per timestep.
- **NDVI** (Rouse et al., 1974): Computed from the red (B4) and near-infrared (B8) Sentinel-2 bands: 1 real-valued dynamic value per timestep.
- **Dynamic World Land Cover classes** (DW, Brown et al., 2022): Land cover classes produced for every non-cloudy Sentinel-2 image: 1 dynamic categorical value from the set of possible classes $\mathcal{V}$ per timestep. We took the mode of classes for all timesteps within a month.
- **Topography data** (TG), from the Shuttle Radar Topography Mission's Digital Elevation Model: The elevation and slope of each pixel, real-valued and static in time.
- **Coordinates** (Loc): 3D static in time Cartesian coordinates computed from the latitude and longitude of the pixel's geographical location: $s_{\text{Loc}} = [\cos(\text{lat}) \times \cos(\text{lon}), \cos(\text{lat}) \times \sin(\text{lon}), \sin(\text{lat})]$.

Hence, one pre-training sample x, comprising a pixel-timeseries $t \in [\mathbb{R}^{T \times 15}; \mathcal{V}^{T \times 1}]$ and static variables $s \in \mathbb{R}^{1 \times 5}$, is summarized as follows. A "pixel-timeseries" refers to both the dynamic and the static variables.

$$x = \left[\left\{ t_i^{\text{S1}};\ t_i^{\text{S2}};\ t_i^{\text{ERA5}};\ t_i^{\text{NDVI}};\ t_i^{\text{DW}} \mid i = 1, ..., 12 \right\};\ s^{\text{TG}};\ s^{\text{Loc}} \right] \tag{1}$$

A.1.2 Channel Groups

As described in Section 2, we transform the pixel timeseries x into a number of tokens, where each token is a linear transformation of a subset of the input channels. We group together channels which (i) come from the same sensor or product, (ii) have equivalent native spatial resolutions and (iii) represent similar parts of the electromagnetic spectrum (for Sentinel-2 channel groups). We group the input data into the following channel groups:

- **Sentinel-1**: The VV and VH bands from the Sentinel-1 sensor
- **Sentinel-2 RGB**: The B2, B3 and B4 bands from the Sentinel-2 sensor
- **Sentinel-2 Red Edge**: The B5, B6 and B7 bands from the Sentinel-2 sensor
- **Sentinel-2 Near Infra Red (10m)**: The B8 band from the Sentinel-2 sensor
- **Sentinel-2 Near Infra Red (20m)**: The B8A band from the Sentinel-2 sensor
- **Sentinel-2 Short Wave Infra Red**: The B11 and B12 bands from the Sentinel-2 sensor
- **NDVI**: The normalized difference vegetation index, calculated from the Sentinel-2 B4 and B8 bands.
- **ERA5 Climatology**: Precipitation and temperature at 2m from the ERA5 Climate Reanalysis product
- **Topography**: The elevation and slope of a pixel, calculated by the SRTM's DEM
- **Location**: The cartesian coordinates of a pixel, computed from the pixel's latitude and longitude

A.2 FLOP calculations

We use the `thop` library (`https://github.com/Lyken17/pytorch-OpCounter`) to calculate the FLOPs required to encode a EuroSAT image (as plotted in Table 2(b)). For the SatMAE, ScaleMAE and ConvMAE models, all images were resized to 224×224, so the FLOPs required to encode an image is independent of resolution. For Presto, we computed the FLOPs required to encode a single pixel and multiplied this by the number of pixels in an image at each resolution (e.g. the "64" resolution has 64×64 pixels, so we multiply the FLOPs required to encode a single pixel by $64 \times 64 = 4096$). The FLOPs calculated by the `thop` library are recorded in Table 4.

A.3 Downstream Results

In addition to the evaluation tasks descrbed in the main paper, we evaluate Presto on 3 additional downstream tasks:

- **Fuel Moisture (timeseries)**: The live fuel moisture dataset (Rao et al., 2020) measures live fuel moisture content in the Western U.S. Rao et al. (2020) baseline used 5-fold cross validation to evaluate model performance; for future comparability, we use a geographically partitioned test set.
- **Algae Blooms (timeseries)**: The algae blooms dataset (alg, 2023) measures the severity of cyanobacterial algal blooms in different parts of the U.S. (we use the subset in the Midwestern U.S.). The dataset was originally released as part of a competition, so the test data is not available. In addition, competitors could download a range of Earth observation datasets to train their models, making direct comparisons to competition results difficult. We benchmark against a regression and a random forest (since the winning solution used a tree-based method), and use a geographically partitioned test set.
- **TreeSatAI (images)**: The TreeSatAI dataset consists of detecting the presence of one or more tree species (out of 20 possible species) in forestry images in Germany (Ahlswede et al., 2023). We use the train and test splits provided by Ahlswede et al. (2023), and compare Presto to the deep learning and tree-based baselines provided alongside the dataset. As done for the baselines, we measure the effectiveness of models using only Sentinel-2 or only Sentinel-1 data.

We include complete results for these additional evaluation tasks. These include error bars, as well as additional results reported for the CropHarvest (Table 5), EuroSAT (Tables 6 and 7) and TreeSatAI datasets (Table 8).

We run all downstream classifiers with 3 seeds $(0, 42, 84)$, with the exception of the KNN classifiers and the linear regression (which are deterministic). In the Table 1 in the main paper, we report the average of these runs; the standard error is reported in Tables 5,8 and 9.

- **Presto as a feature extractor**: When used as a feature extractor, a random forest, regression of K-nearest-neighbours classifier is trained on Presto's output embeddings. In this case, we use scikit-learn models with the default hyperparameters. The CropHarvest tasks, the class labels are extremely balanced; we therefore set `class_weight` equal to `balanced` for those tasks, for both Presto and MOSAIKS-1D.
- **Fine-tuning Presto**: When fine-tuning Presto, we use the same hyperparameters across all tasks: an AdamW optimizer with a learning rate of `3e-4` and a batch size of 64. We use a geographically seperated validation set with early stopping, with a patience of 10.

For image-based downstream tasks, we obtain per-image predictions using Presto by computing a mean and standard deviation of Presto's output pixels, and passing a concatenation of these two vectors to a downstream classifier. This is illustrated in Figure 4.

Table 5: Additional results for the CropHarvest task. In addition to the F1 scores reported in the main paper, we report AUC ROC scores. In addition, we report error bars computed with three runs (for all models).

	Model	Kenya	Brazil	Togo	Mean
F1	Random Forest	0.559 ± 0.003	0.000 ± 0.000	0.756 ± 0.002	0.441
	MOSAIKS-1D$_R$	0.790 ± 0.027	0.746 ± 0.084	0.679 ± 0.024	0.738
	TIML	0.838 ± 0.000	0.835 ± 0.012	0.732 ± 0.002	0.802
	Presto$_R$	0.816 ± 0.000	$\mathbf{0.891 \pm 0.000}$	$\mathbf{0.798 \pm 0.000}$	0.835
	no DW	$\mathbf{0.861 \pm 0.000}$	0.888 ± 0.000	0.760 ± 0.000	**0.836**
AUC ROC	Random Forest	0.578 ± 0.006	0.941 ± 0.004	0.892 ± 0.001	0.803
	MOSAIKS-1D$_R$	0.693 ± 0.036	0.890 ± 0.038	0.836 ± 0.005	0.806
	TIML	0.794 ± 0.003	0.988 ± 0.001	0.890 ± 0.000	0.890
	Presto$_R$	0.834 ± 0.000	$\mathbf{0.997 \pm 0.000}$	$\mathbf{0.921 \pm 0.000}$	0.917
	no DW	$\mathbf{0.863 \pm 0.000}$	0.989 ± 0.000	0.912 ± 0.000	**0.921**

Table 6: Additional results for the EuroSat task - results for the ScaleMAE, SatMAE and ConvMAE models are from Reed et al. (2022). We report KNN classifier results for different values of k, and at varying input resolutions.

Resolution	16			32			64		
k	5	20	100	5	20	100	5	20	100
SatMAE	0.729	0.727	0.695	0.871	0.876	0.854	0.934	0.931	0.913
ScaleMAE	0.751	0.744	0.699	0.912	0.901	0.869	**0.960**	0.956	0.935
ConvMAE	0.835	0.826	0.788	0.909	0.898	0.863	0.947	0.940	0.914
Presto (RGB)	0.869	0.828	0.713	0.869	0.829	0.712	0.869	0.829	0.713
Presto (MS)	**0.916**	0.892	0.844	**0.920**	0.892	0.846	0.921	0.893	0.846

Table 7: Additional results for the EuroSat task for Presto when run with reduced resolutions (compared to those used by Reed et al. (2022) and reported in Table 6). We report KNN classifier results for different values of k, and at varying input resolutions.

Resolution	2			4			8		
k	5	20	100	5	20	100	5	20	100
Presto (RGB)	0.843	0.811	0.699	0.860	0.820	0.706	0.869	0.826	0.710
Presto (MS)	0.873	0.852	0.799	0.895	0.874	0.824	0.911	0.886	0.838

Table 8: Results on the TreeSatAI dataset. We compare Presto to the dataset's benchmark models. The MLPs contain 3 layers (with 563K-723K parameters respectively) and are tuned for this task, whereas we freeze the Presto encoder's 401k parameters and train a random forest on its outputs with default scikit-learn hyperparameters.

Model	Data	Aggregation	F_1	mAP	Precision	Recall
MLP	S1	Weighted	10.09	29.42	33.29	7.13
LightGBM			11.86	32.79	37.96	8.06
Presto$_{RF}$			19.79 ± 0.00	35.76 ± 0.00	51.90 ± 0.02	14.16 ± 0.00
MLP		Micro	12.82	33.09	63.01	7.13
LightGBM			14.07	35.11	55.49	8.06
Presto$_{RF}$			22.92 ± 0.00	38.69 ± 0.00	60.17 ± 0.00	14.16 ± 0.00
MLP	S2	Weighted	51.97	64.19	74.59	42.23
LightGBM			48.17	61.99	74.27	40.04
Presto$_{RF}$			46.26 ± 0.00	60.88 ± 0.00	75.42 ± 0.00	37.08 ± 0.00
MLP		Micro	54.49	65.83	77.18	42.23
LightGBM			52.52	61.66	76.27	40.04
Presto$_{RF}$			50.41 ± 0.00	63.24 ± 0.00	78.70 ± 0.00	37.08 ± 0.00

Table 9: RMSE results on the regression tasks. While the literature baselines are not directly comparable, since they use different input datasets or private test data (or both), Rao et al. (2020) report an RMSE of 25 on the fuel moisture dataset with a physics-assisted neural network and the algae bloom competition winner reported an RMSE of 0.761, indicating our results are within the scope of utility. Best results are **highlighted blue**, with second best results in **bold**. To account for random seeding, results are an average of three runs with standard error reported. Models have a high variance in performance across tasks – we therefore calculate the mean difference in RMSE from the linear regression baseline across both tasks. Presto performs most consistently, both when used as a feature-extractor for random forests and when fine-tuned.

	Fuel Moisture	Algae Blooms	Mean difference
Linear Regression	28.20	**0.850**	0%
Random Forest	**23.84 ± 0.42**	1.249 ± 0.02	15.7%
MOSAIKS-1D$_{RF}$	28.75 ± 0.15	0.972 ± 0.01	8.15%
Fully Supervised	26.07 ± 0.52	0.955 ± 0.05	2.40%
Presto$_{FT}$	**25.28 ± 0.30**	**0.815 ± 0.03**	**−7.24%**
Presto$_{RF}$	25.98 ± 0.66	0.884 ± 0.01	**−1.94%**

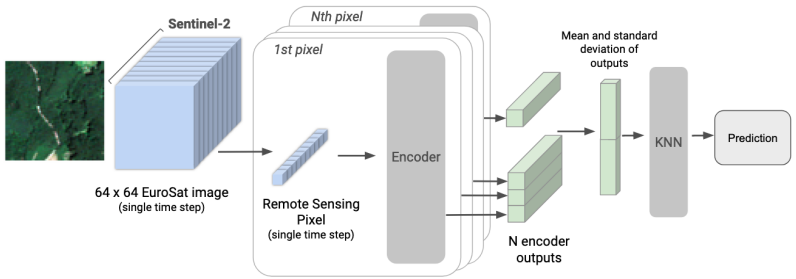


Figure 4: We obtain per-image predictions using Presto by computing a mean and standard deviation of Presto's per-pixel outputs, and passing this concatenated vector to a downstream classifier. We illustrate what this process looks like for the EuroSat task.