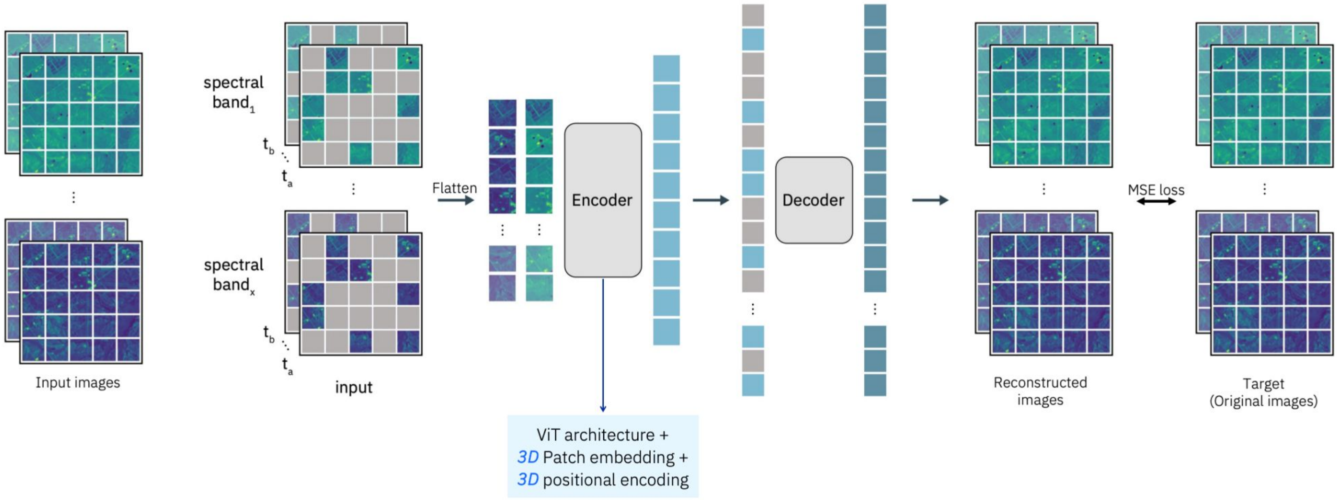
Foundation Models for Generalist Geospatial



Ryan Chesler
San Diego Machine Learning

Abstract

- NASA/IBM geospatial foundation model
- Large scale pre-training of a masked image model for satellite images
- Technical challenges with this size of data
- Model can transfer knowledge to new problems

Pre-training Data

- HLS-2
  - 30 meter resolution
  - Temporal resolution of every 2-3 days
  - Harmonized from multiple satellites
  - 3660x3660 tiles with 15 spectral bands
  - Goes back to 2015
  - 3.61 Petabytes of data
- Clustered the data into 20 zones and sampled

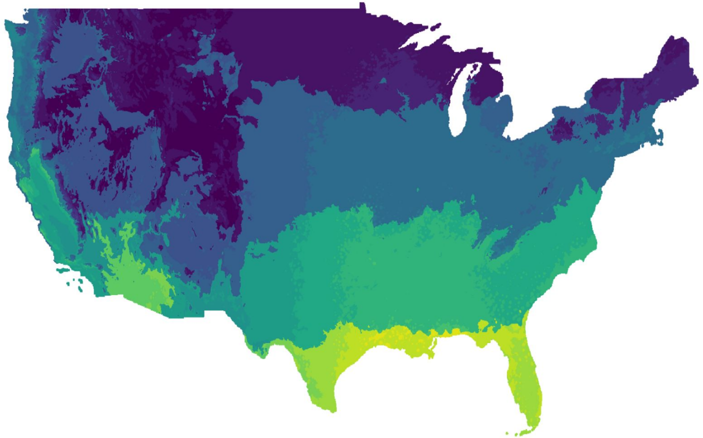


Fig. 2: Geo-regions from the contiguous U.S. are clustered into one of 20 different categories based on temperature and precipitation data.

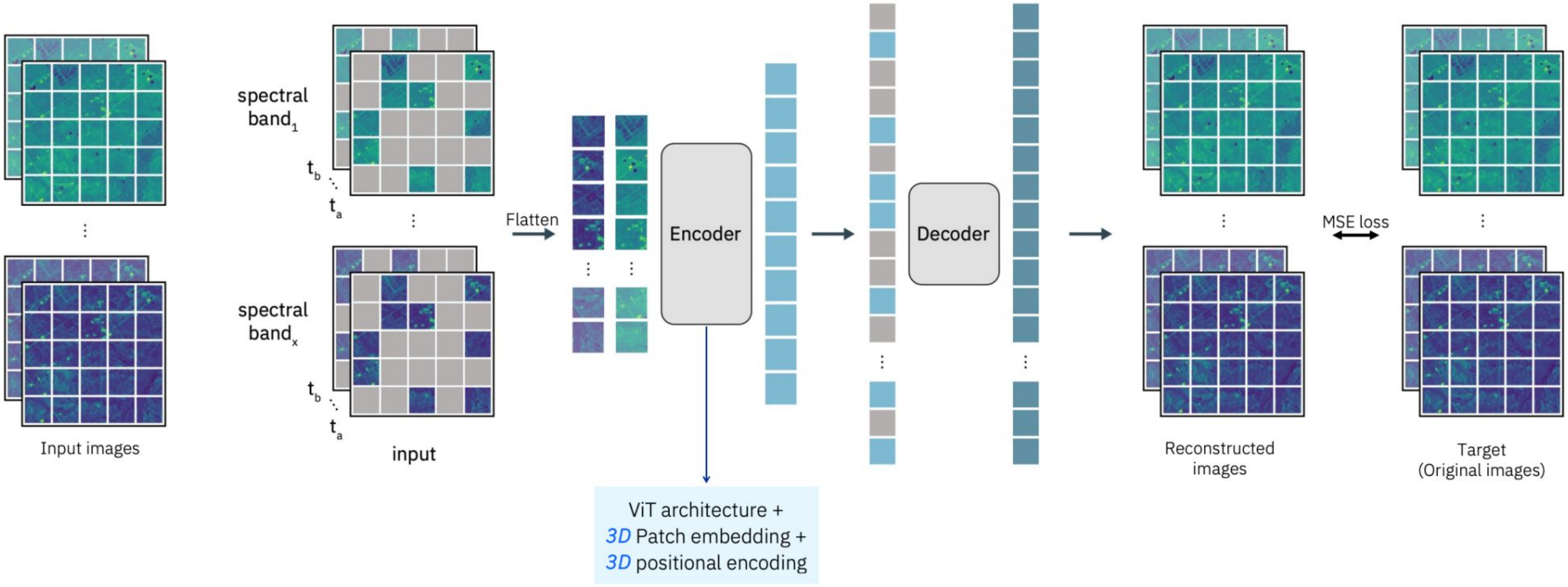
Reformatting to Zarr

- Data had to be filtered offline before training
- Preprocessing GeoTIFF files during training too slow
- Original files - 3660 x 3660 x 15 x 3
- Model input - 224 x 224 x 6 x 3
- 667x more reading than is necessary
- Further exacerbated by needing to filter missing and cloud covered areas
- Precomputed valid regions to sample from

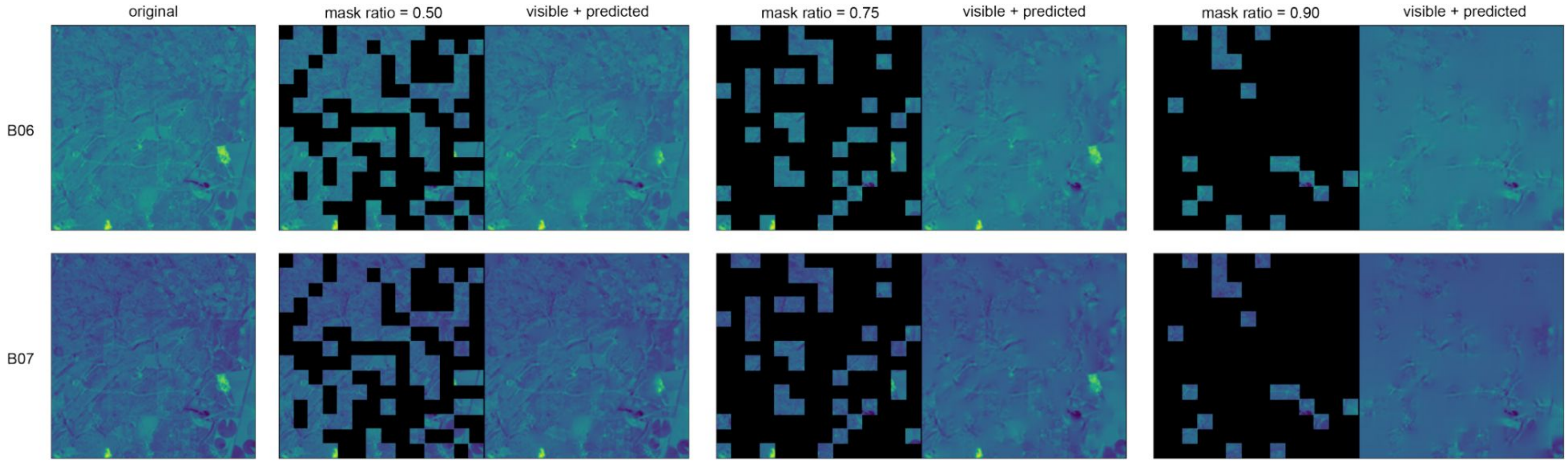
	batch/GPU	workers	prefetch	epoch avg time (s)
GeoTiff 64 GPUs	16	1	2	384
GeoTiff 8 GPUs	128	8	2	690
Zarr 8 GPUs	128	2	4	**381**

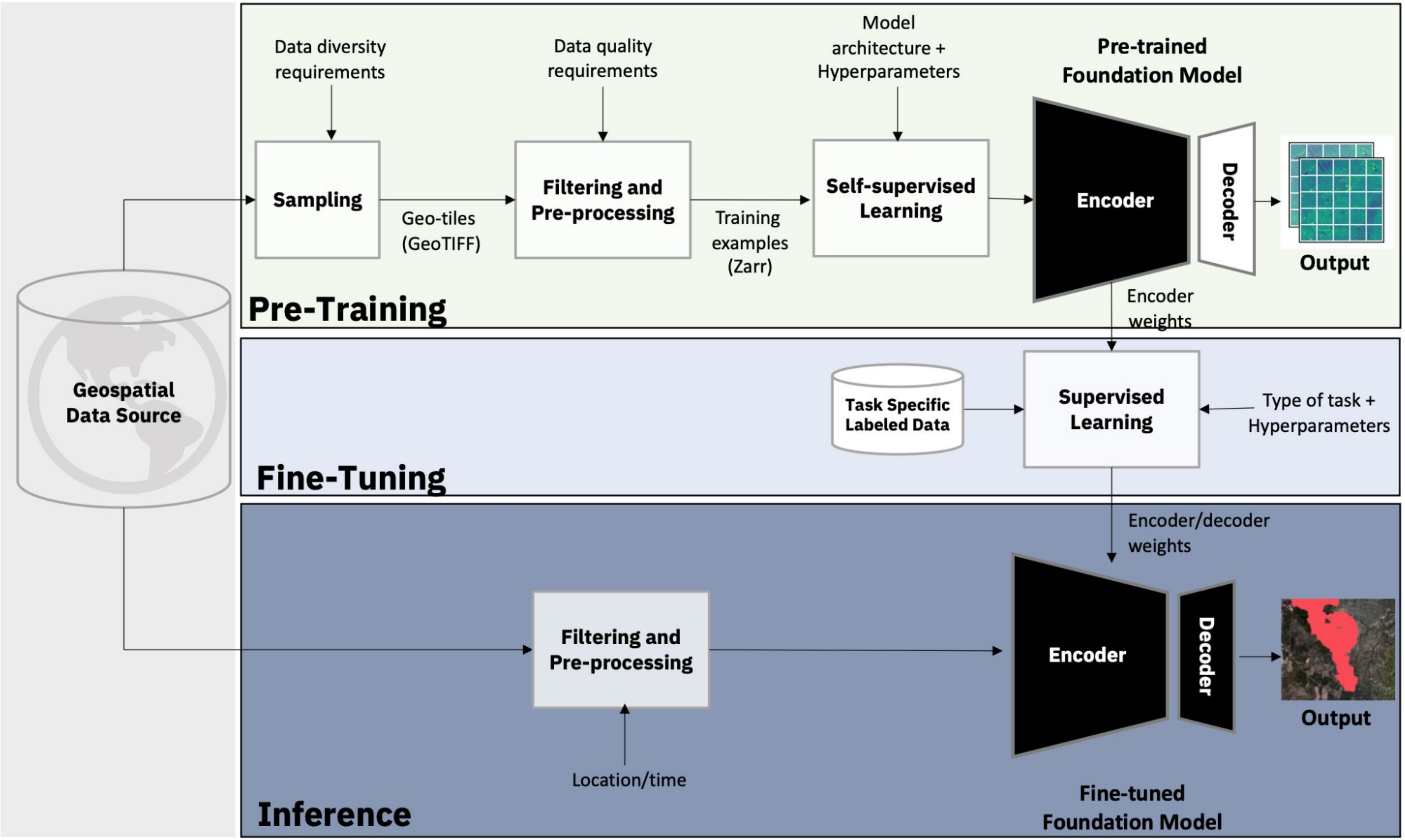
Table 1: Average epoch time in seconds for different runs of data preprocessing and loading. Zarr-based data loading is approximately two times faster than corresponding GeoTiff loading.

Architecture/Training Process



Masked autoencoding



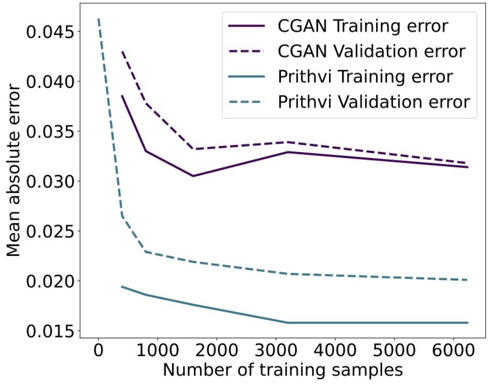


Pre-Training
Data diversity requirements
Data quality requirements
Model architecture + Hyperparameters
Pre-trained Foundation Model
Sampling
Geo-tiles (GeoTIFF)
Filtering and Pre-processing
Training examples (Zarr)
Self-supervised Learning
Encoder
Decoder
Output
Encoder weights
Geospatial Data Source
Fine-Tuning
Task Specific Labeled Data
Supervised Learning
Type of task + Hyperparameters
Encoder/decoder weights
Inference
Filtering and Pre-processing
Location/time
Encoder
Decoder
Output
Fine-tuned Foundation Model

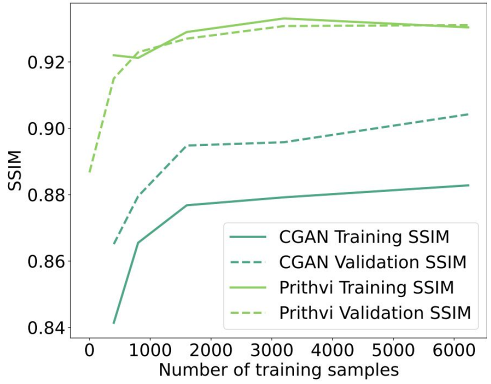
Downstream tasks

- Multi-Temporal Cloud Gap Imputation
- Flood Mapping
- Wildfire Scar Mapping
- Multi-Temporal Crop Segmentation

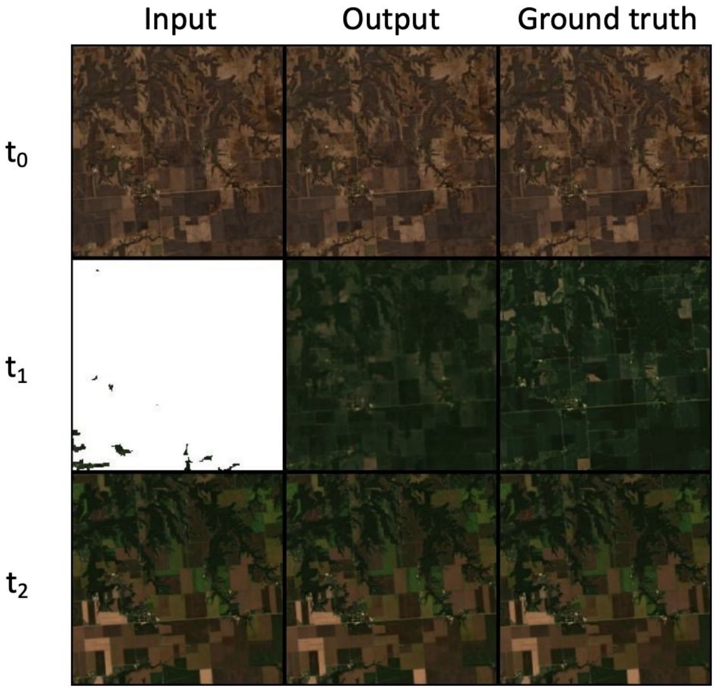
Multi-temporal cloud gap filling



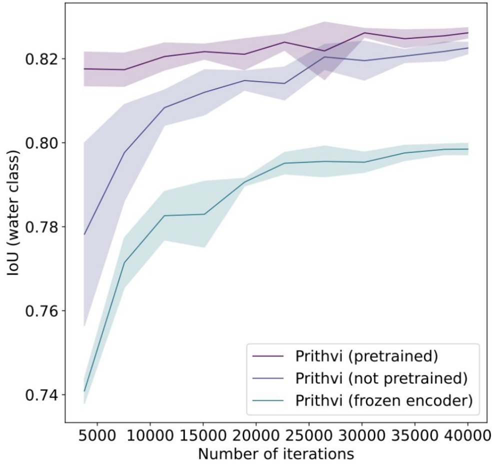
(a) Mean absolute error after 200 epochs.

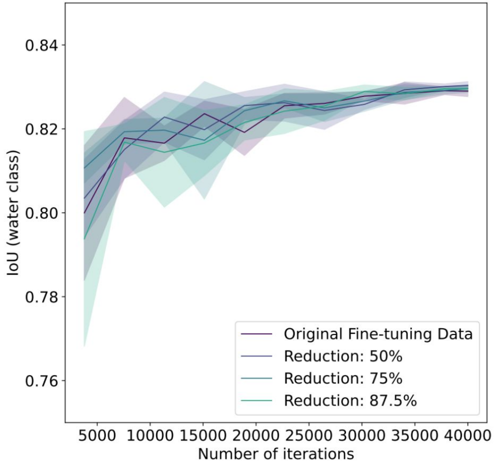


(b) Structural similarity index measure afte
200 epochs.



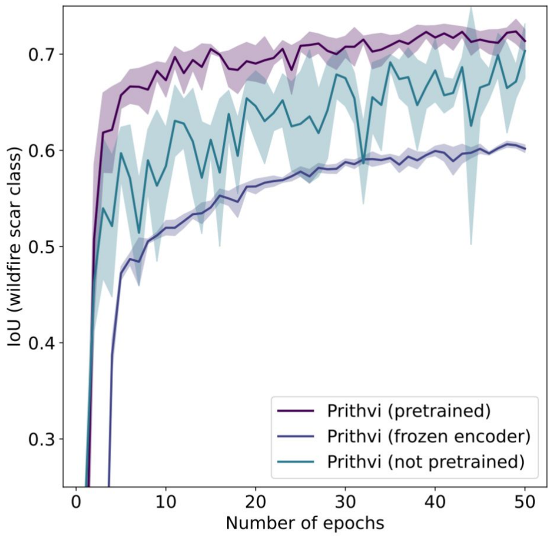
Flood mapping

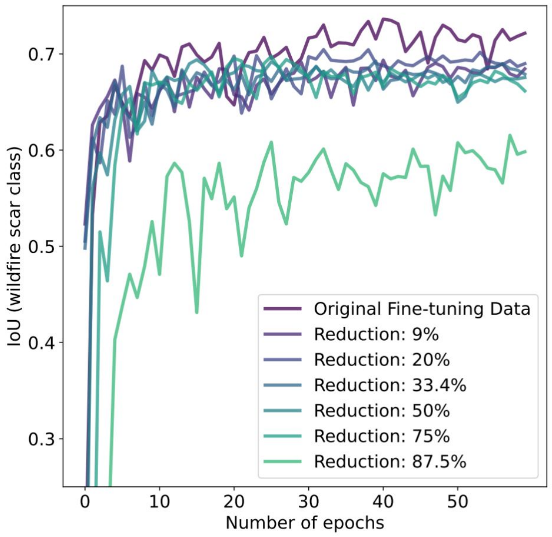




	IoU (water)	F1 (water)	mIoU (both classes)	mF1-score (both classes)	mAcc (both classes)
Baseline [55]	24.21	–	–	–	–
ViT-base [19]	67.58	80.65	81.06	88.92	88.82
Swin [60]	79.43	88.54	87.48	93.13	90.63
Swin† [60]	80.58	89.24	87.98	93.44	92.02
AFTER 50 EPOCHS					
Prithvi (not pretrained)	80.67	89.30	88.76	93.85	94.79
Prithvi (pretrained)	81.26	89.66	89.10	94.05	**95.07**
AFTER 500 EPOCHS					
Prithvi (not pretrained)	82.97	90.69	90.14	94.66	94.82
Prithvi (pretrained)	**82.99**	**90.71**	**90.16**	**94.68**	94.60

Wildlife scar mapping





	IoU (fire scar)	F1 (fire scar)	mIoU (both classes)	mF1-score (both classes)	mAcc (both classes)
U-Net (DeepLabV3) [61]	71.01	83.05	83.55	90.53	87.98
ViT-base [19]	69.04	81.69	82.20	89.65	90.14
Prithvi (not pretrained)	72.26	83.89	84.01	90.87	92.41
Prithvi (pretrained)	**73.62**	**84.81**	**84.84**	**91.40**	**92.48**

Conclusion

- Different from the previous geospatial foundational model we looked at
  - Filling in image instead of forecasting
  - Focused on individual data source
- Performance is good but not paradigm shifting