Multimodal Foundation Model

SAIR-2-09 :Comment and Discuss "Foundation Models for Generalist Geospatial Artificial Intelligence

Momiao Xiong

Society of Artificial Intelligence Research

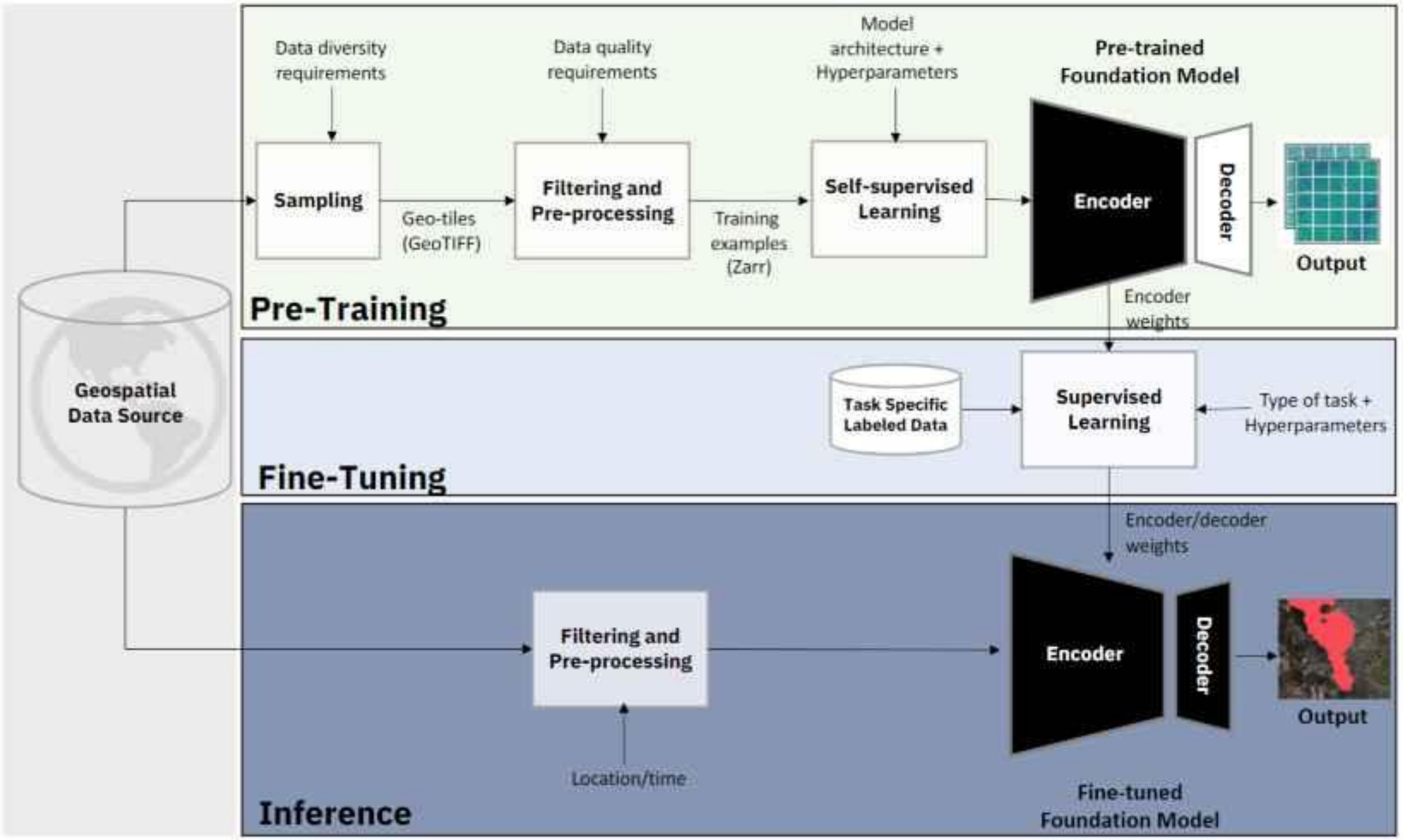


Fig. 1: We propose a first-of-its-kind framework for the development of geospatial foundation models from raw satellite imagery, which we leverage to generate the Prithvi-100M model. The framework encompasses (1) the sampling, filtering, and pre-processing of raw geospatial data and the self-supervised foundation model pretraining, (2) the fine-tuning to specific downstream applications, and (3) the inference process.

Efficient Data Sampling

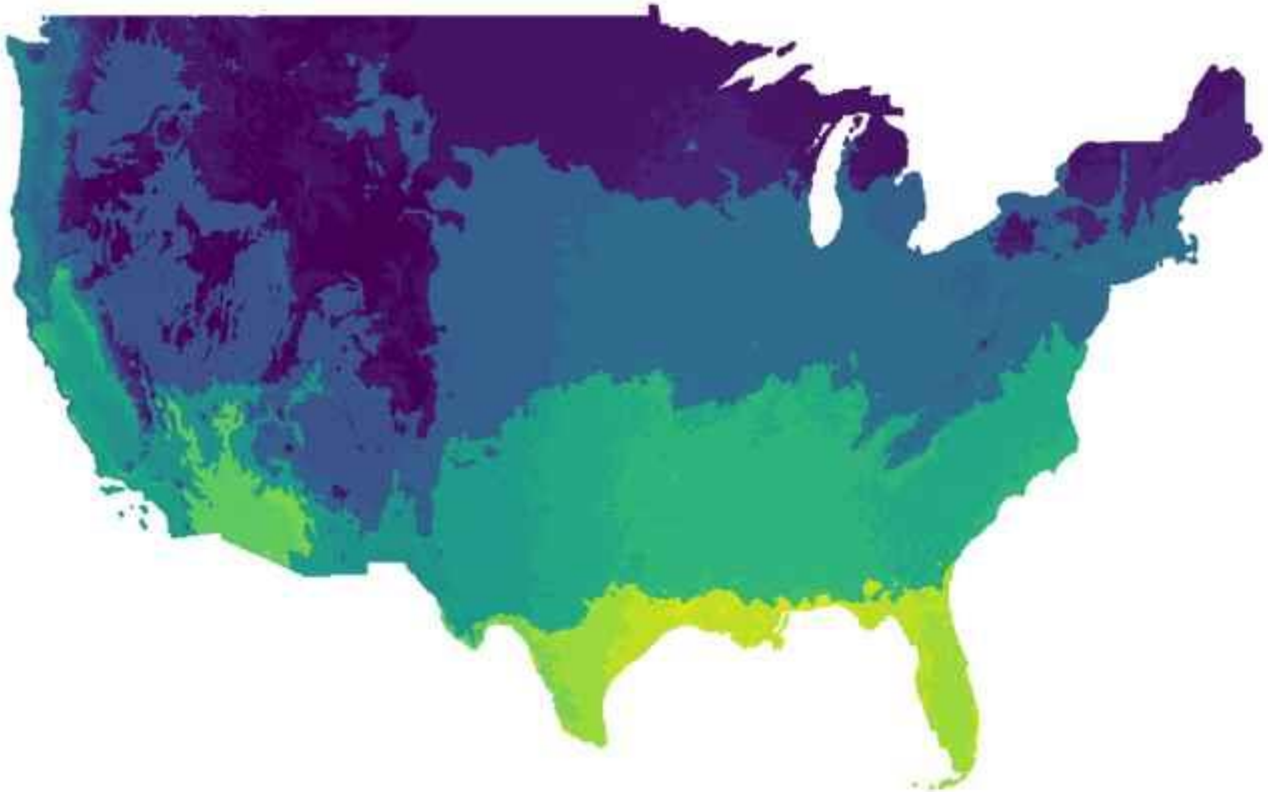
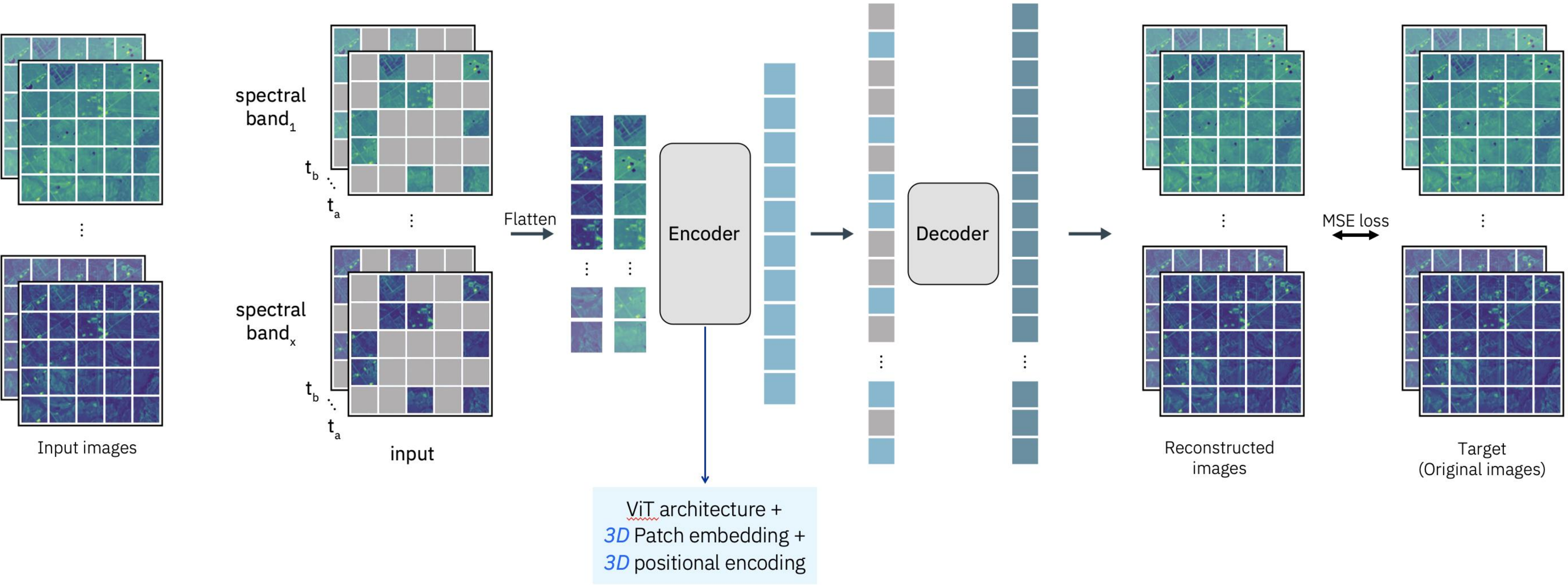


Fig. 2: Geo-regions from the contiguous U.S. are clustered into one of 20 different categories based on temperature and precipitation data.



The masked autoencoder (MAE) structure for pre-training Prithvi on large-scale multi-temporal and multi-spectral satellite images.

Our main modifications to the ViT architecture are the 3D positional embedding and the 3D patch embedding, which are required to deal with the spatiotemporal data.

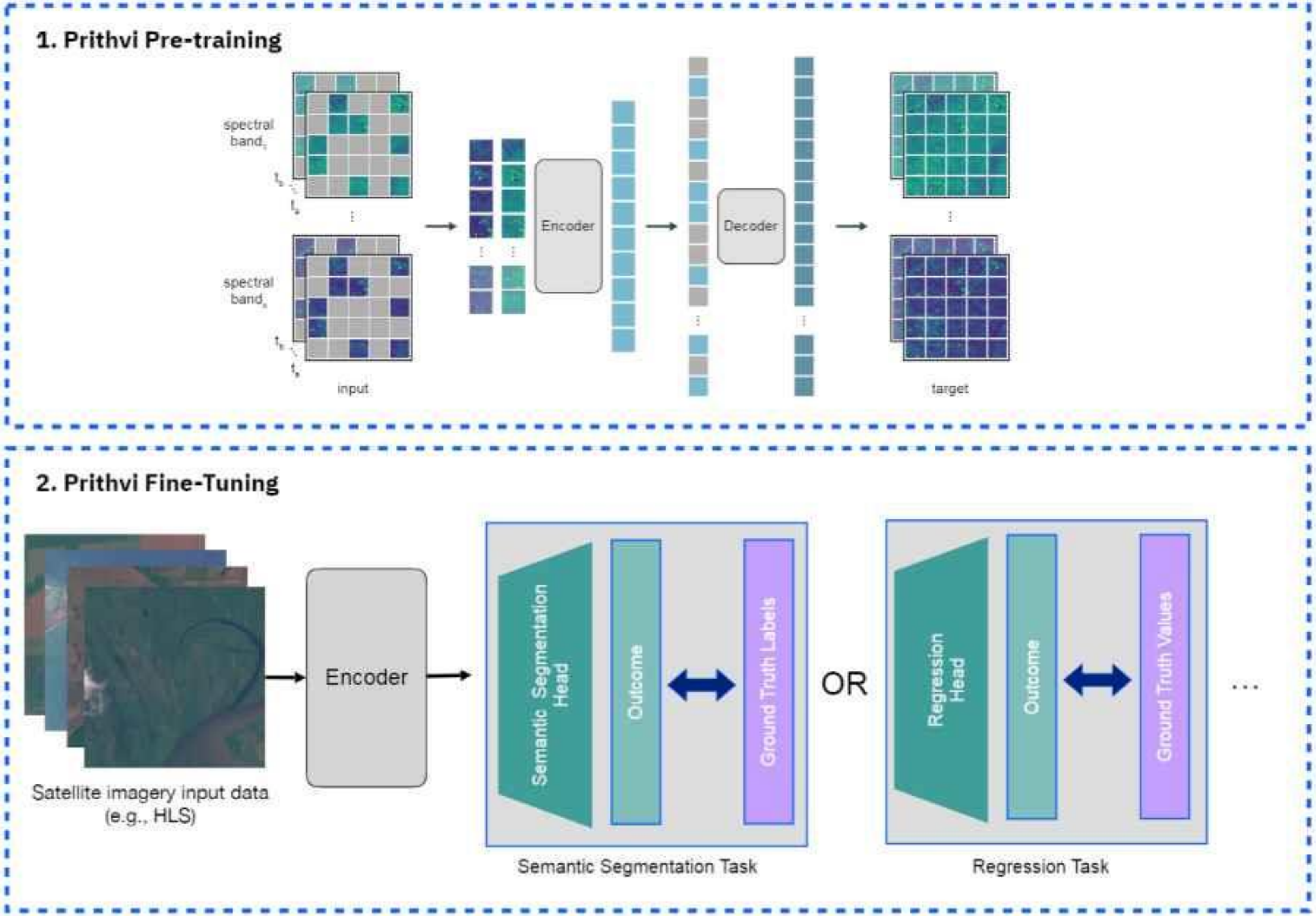
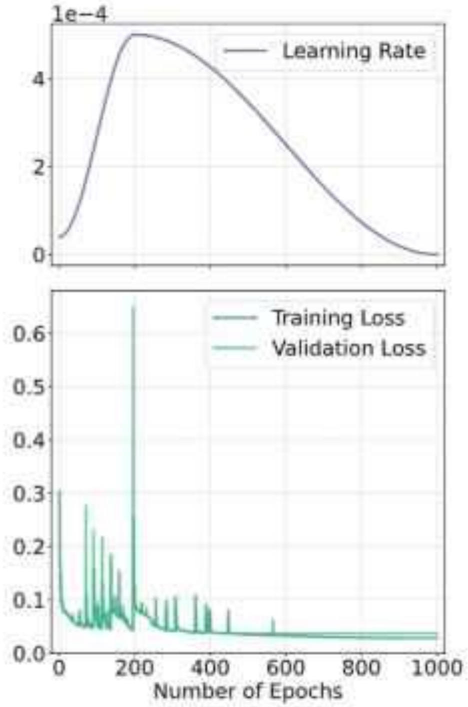
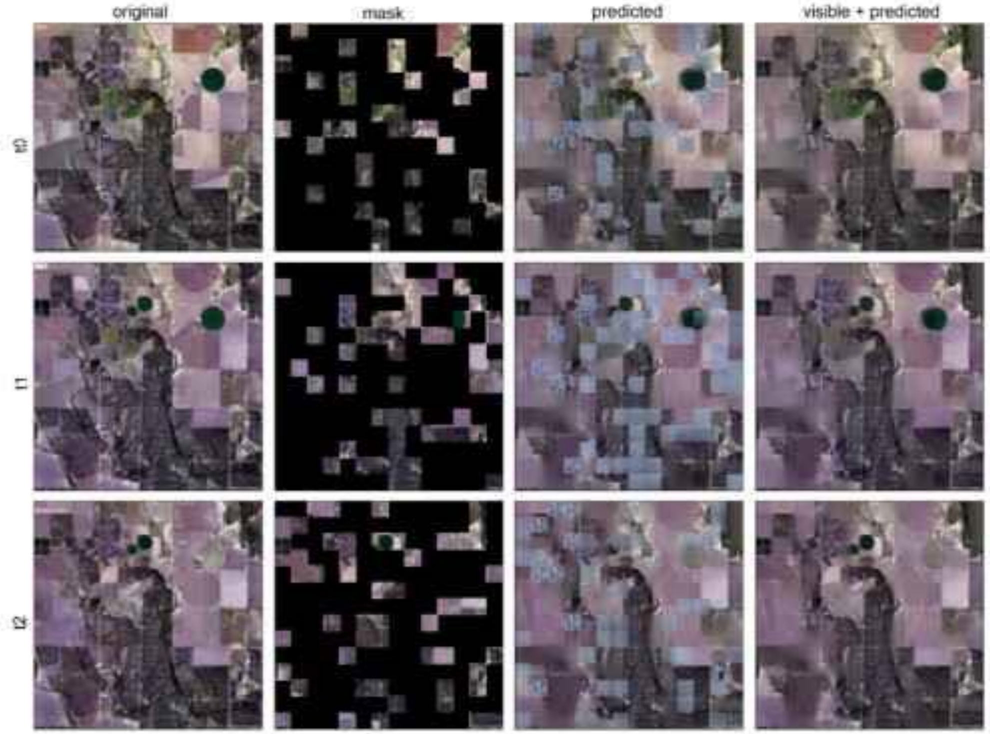


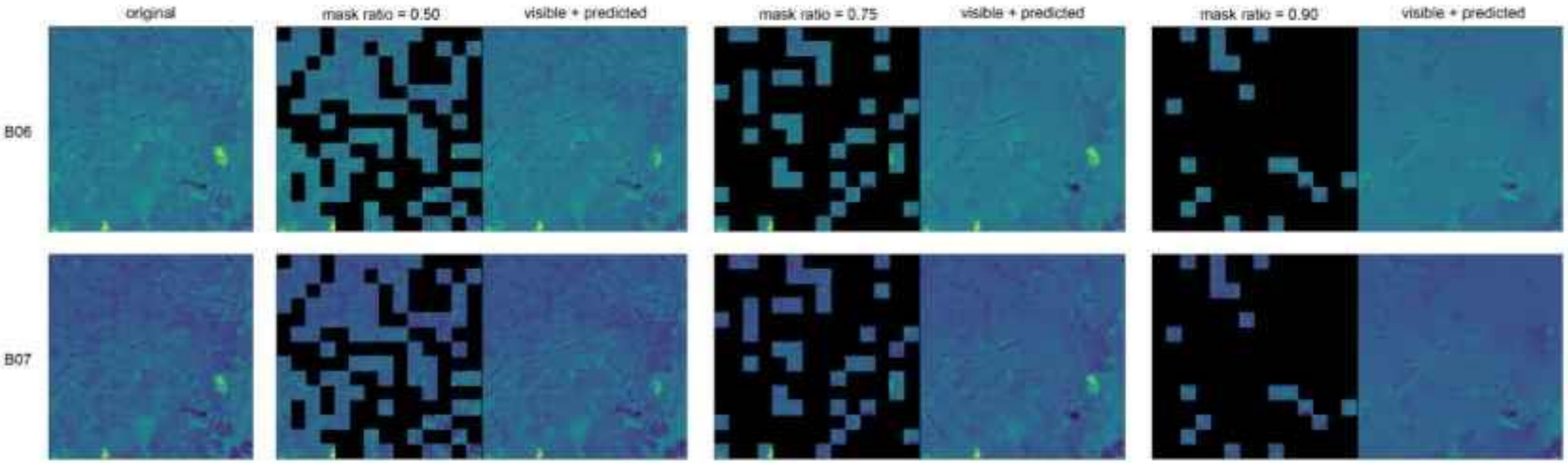
Fig. 4: Pre-training and fine-tuning in Prithvi for various types of downstream tasks.





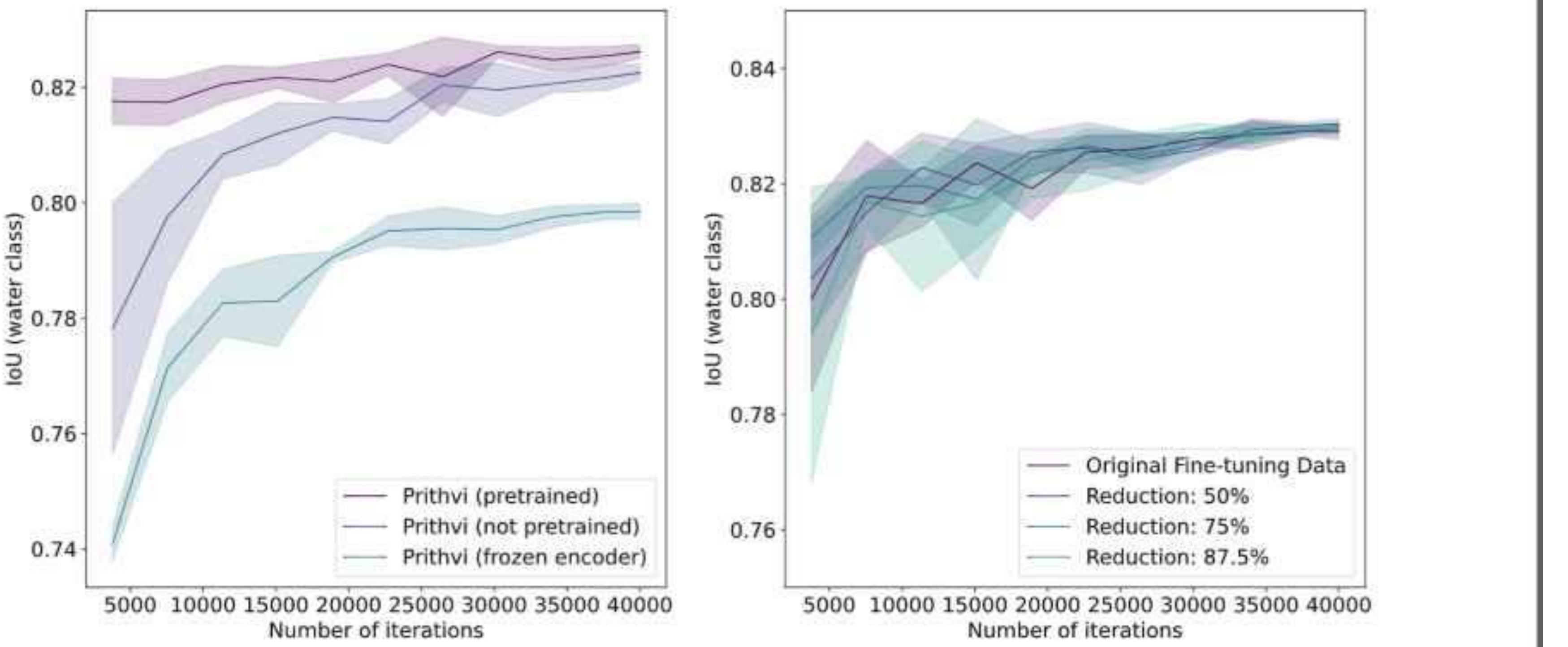
(a) MSE training and validation loss curves during pretraining accompanied by the associated values of the learning rate scheduler. Training loss decreases to 0.0283, validation loss is lowest at 0.0364.

(b) Reconstruction results on images unseen during training (different locations) with Prithvi model with ViT-base backbone. Here we show the RGB bands together (B04, B03, and B02, respectively) for better visualization, although the model also predicts B05, B06, and B07.



(c) Reconstruction results on images unseen during training (different locations) with Prithvi model for bands B06 and B07 for different masking ratios with ViT-base backbone. Here, we show a single time step of an input image unseen during training.

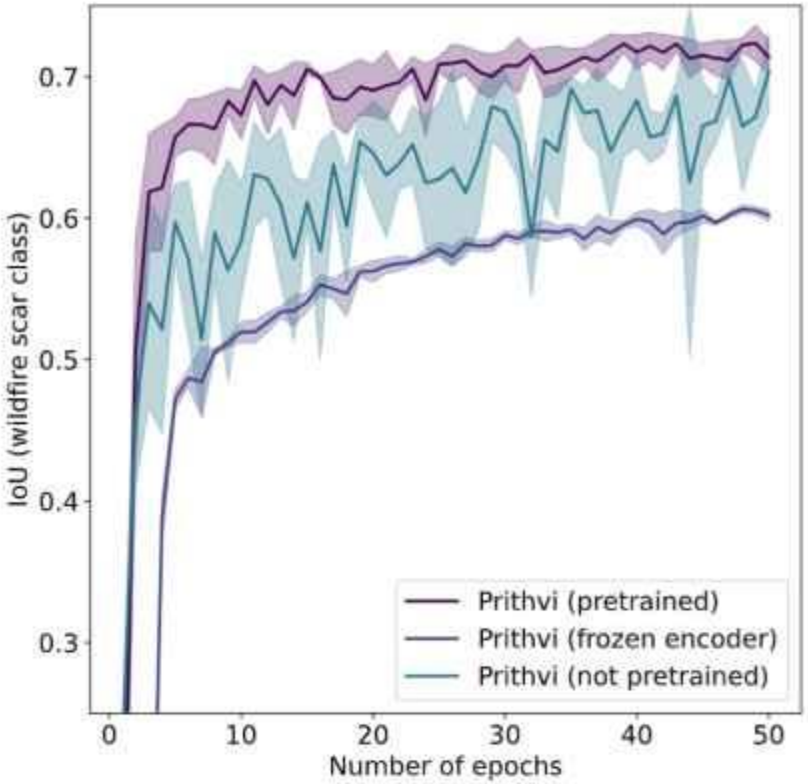
Fig. 5: Pretraining results of Prithvi using 1TB of HLS data from the contiguous US.



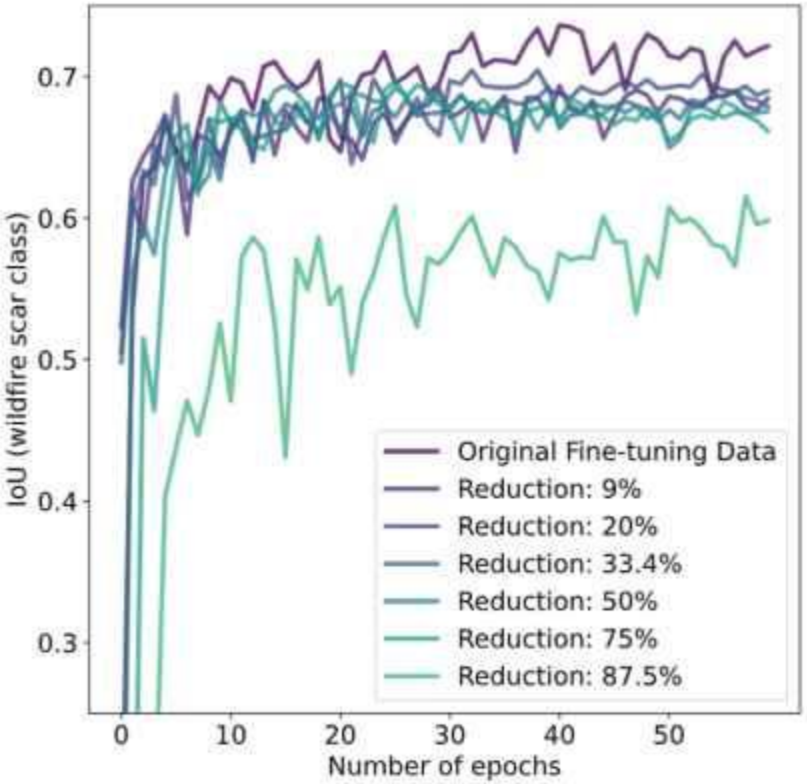
(a) Performance based on (1) pretrained, (2) randomly initialized, and (3) frozen encoder weights. Confidence bands represent the standard deviation across 5 different seeds.

(b) Data efficiency of pretrained Prithvi in terms of reduction of required labeled images for fine-tuning in the flood mapping task using ViT-large backbone.

Fig. 9: Evaluation of Prithvi on Sen1Floods11 test set regarding (a) the performance and (b) the data efficiency using the ViT-large backbone.



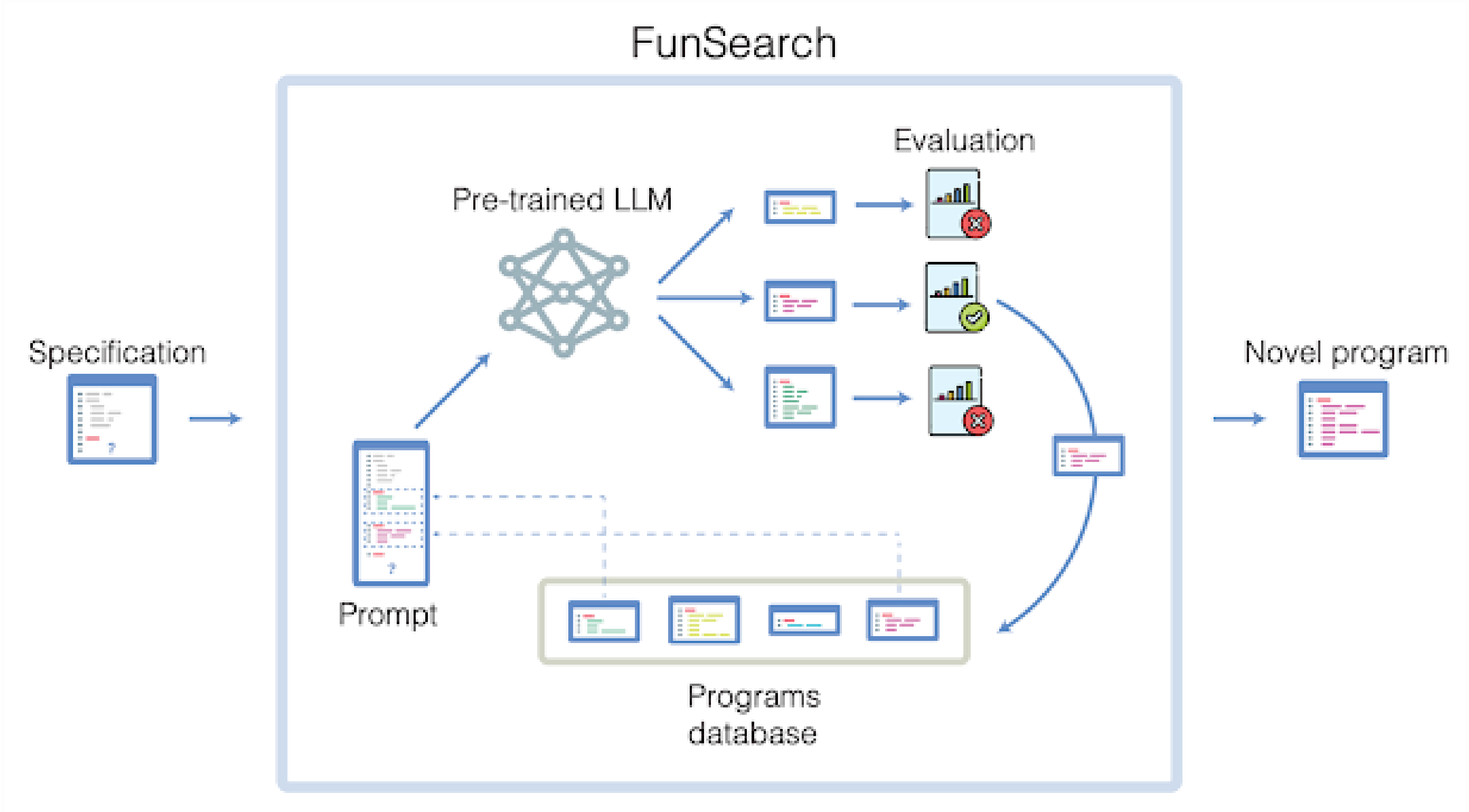
(a) Performance based on (1) pretrained, (2) randomly initialized, and (3) frozen encoder weights. Confidence bands represent the standard deviation across 5 different seeds.



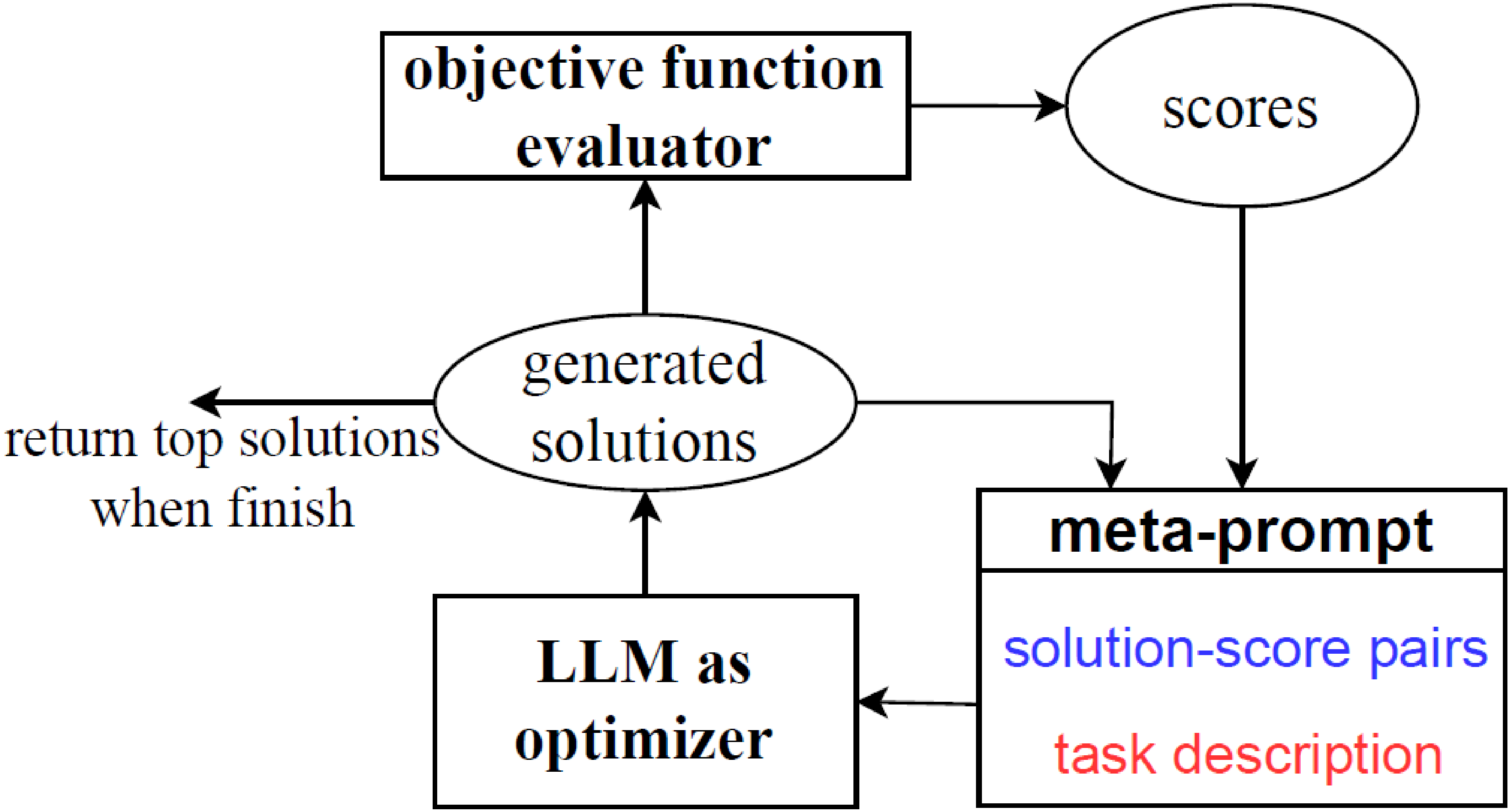
(b) Pre-trained Prithvi: Data efficiency in terms of reduction of required labeled images for fine-tuning in the wildfire scar segmentation task using ViT-base backbone.

Classes	Prithvi		U-Net	
	Accuracy	IoU	Accuracy	IoU
Natural Vegetation	46.89%	0.4038	63.67%	0.4578
Forest	66.38%	0.4747	71.72%	0.4772
Corn	65.47%	0.5491	63.33%	0.5226
Soybeans	67.46%	0.5297	66.77%	0.5168
Wetlands	58.91%	0.4020	60.36%	0.4110
Developed/Barren	56.49%	0.3611	60.23%	0.4637
Open Water	90.37%	0.6804	87.76%	0.7596
Winter Wheat	67.16%	0.4967	66.39%	0.4950
Alfalfa	66.75%	0.3084	59.03%	0.3848
Fallow/Idle Cropland	59.23%	0.3493	52.94%	0.3599
Cotton	66.94%	0.3237	45.30%	0.3258
Sorghum	73.56%	0.3283	61.53%	0.3910
Other	47.12%	0.3427	45.90%	0.3268
Mean	**64.06%**	**0.426**	61.91%	**0.420**

Table 4: Prithvi model performance for the crop segmentation based on three input timestep compared to a U-Net baseline. For this study, Prithvi was fine-tuned on the CDL dataset for 80 epochs with three input time steps, and U-Net was trained for 100 epochs



Mathematical discoveries from program search with large language models



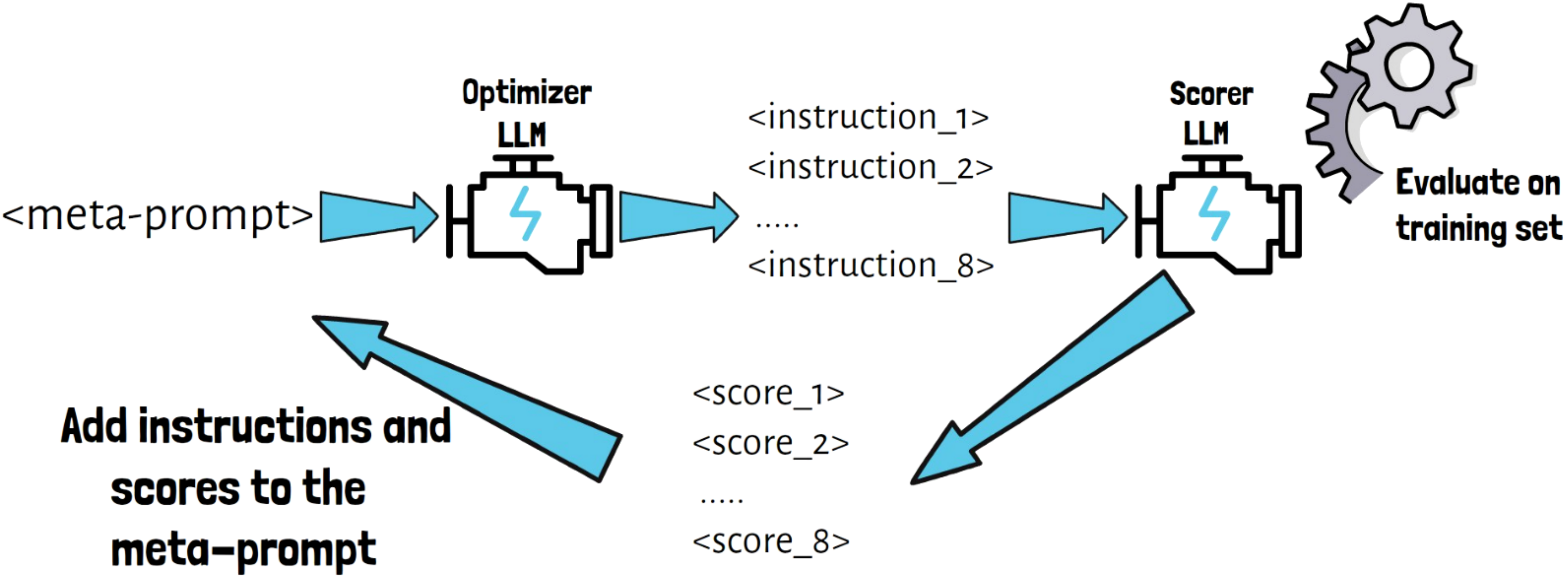
LARGE LANGUAGE MODELS AS OPTIMIZERS

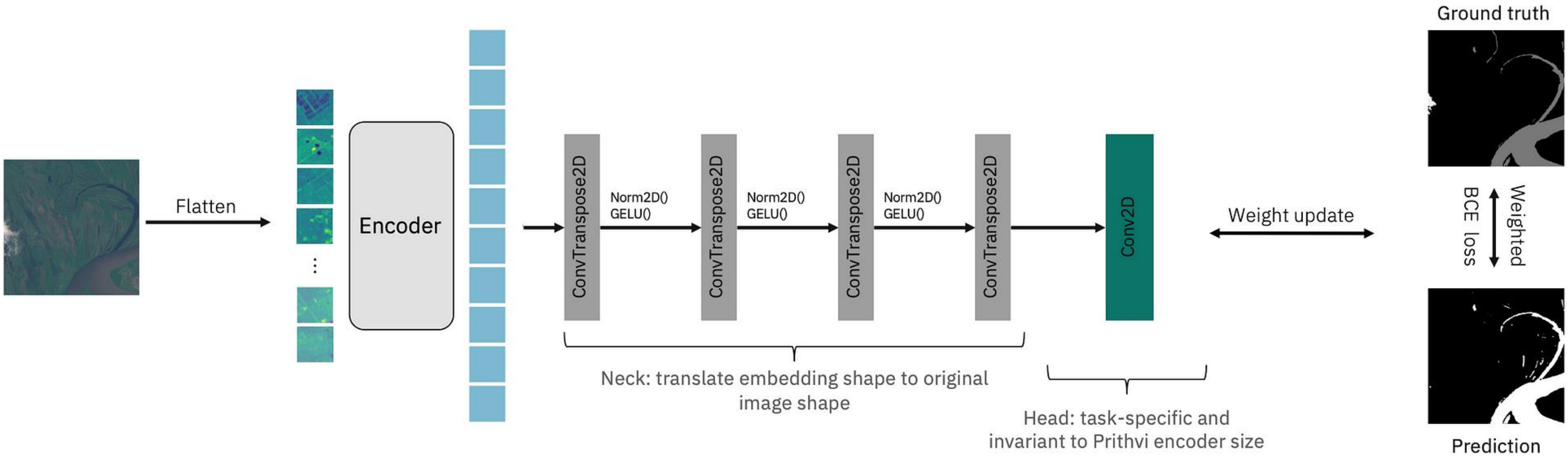
Code at https://github.com/ google-deepmind/opro.

https://aipapersacademy.com/large-language-models-as-optimizers/

Goal – maximize the accuracy over a dataset

How? Create an instruction that will be added to the prompts




Flatten
Encoder
ConvTranspose2D
Norm2D()
GELU()
ConvTranspose2D
Norm2D()
GELU()
ConvTranspose2D
Norm2D()
GELU()
ConvTranspose2D
Conv2D
Neck: translate embedding shape to original image shape
Head: task-specific and invariant to Prithvi encoder size
Weight update
Ground truth
Weighted BCE loss
Prediction