Foundation Models for Earth Observation

Ayush Prasad

Agenda

1. Motivation
2. How we got here
3. Current Geospatial Foundation Models
4. How they’re trained
5. What you can do with them
6. Frozen encoder vs fine-tuning
7. An example use case for sea ice segmentation

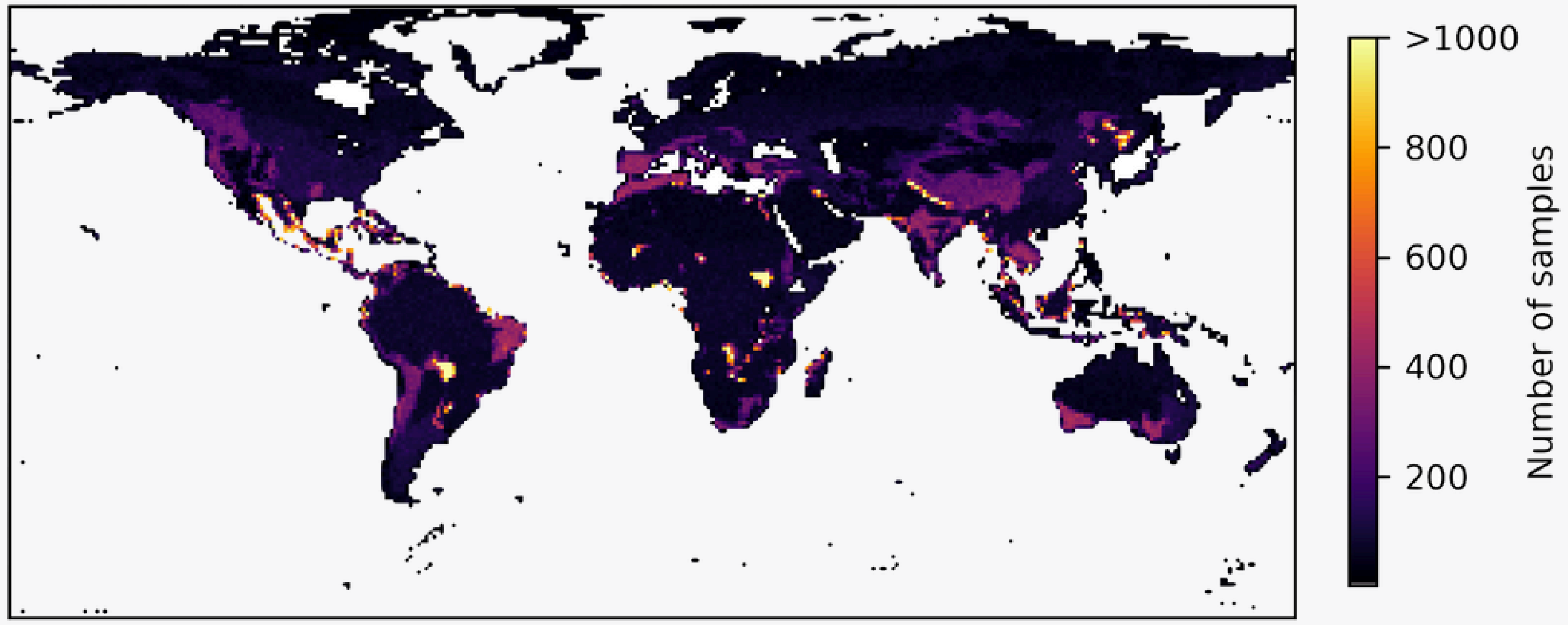
Perform Sea Ice Segmentation with TerraMind

https://github.com/ayushprd/sea_ice_segmentation

Tutorial and Slides

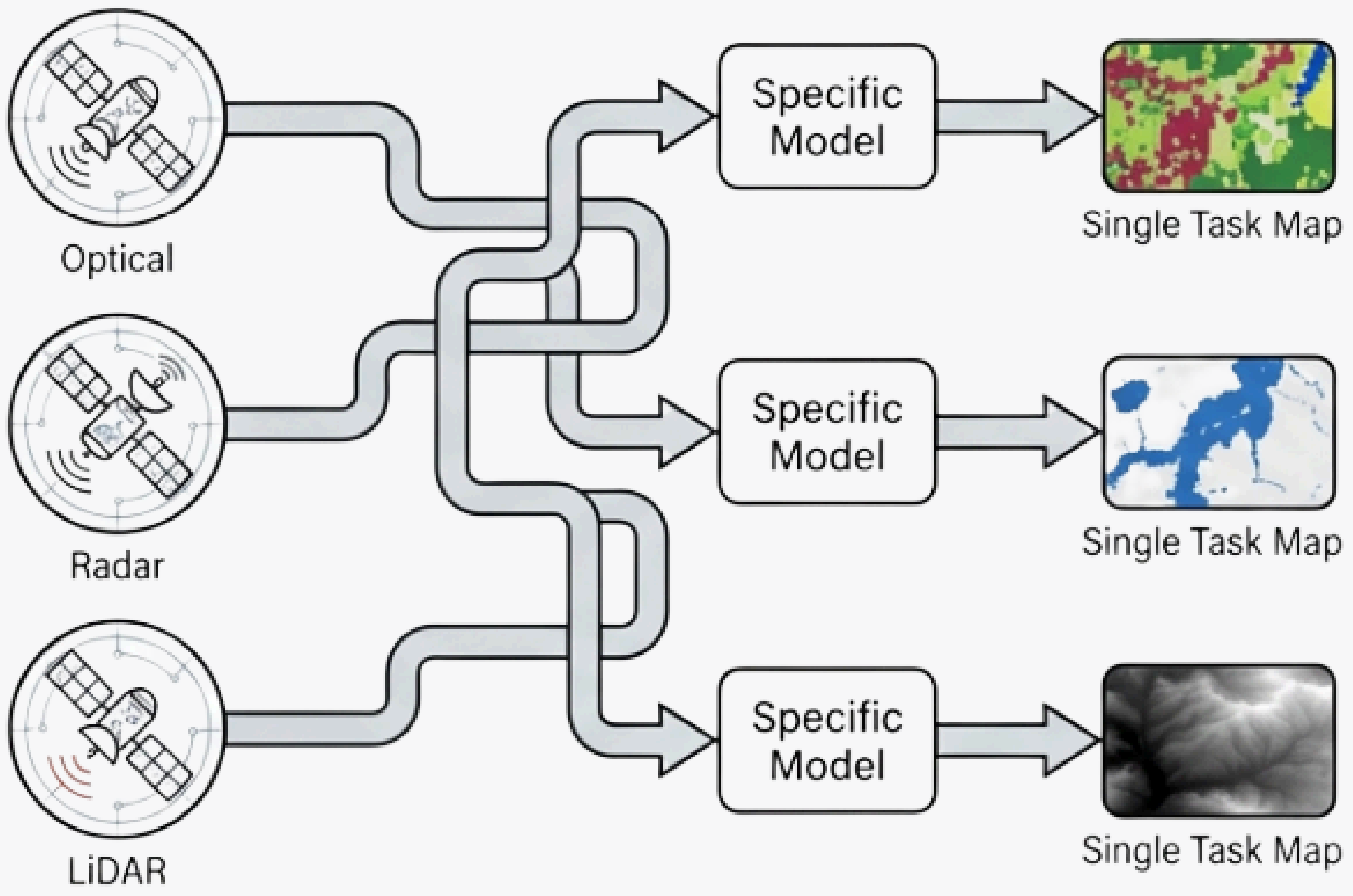
Geospatial Foundation Models

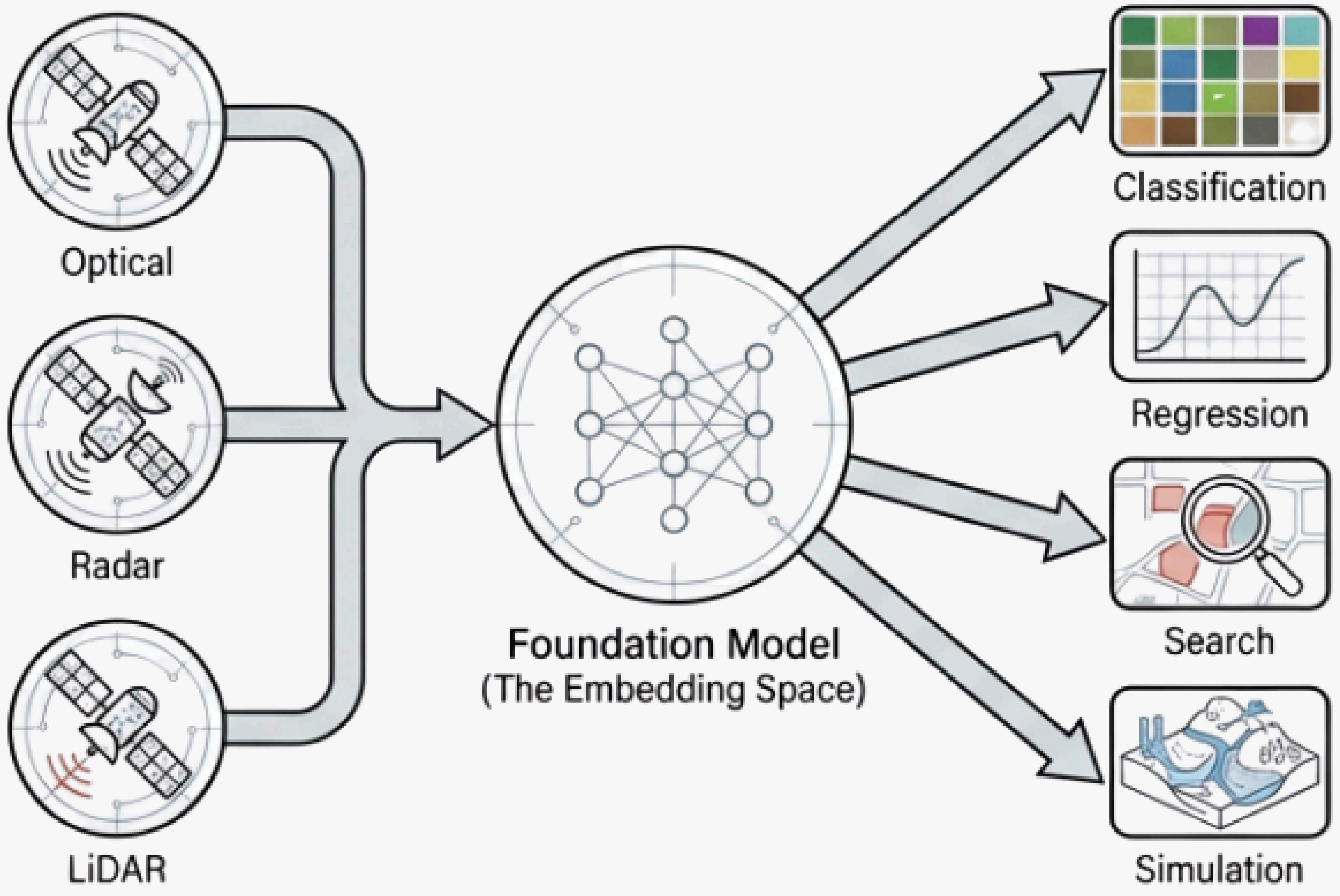
- Large-scale models trained on global heterogeneous data (satellite imagery, weather, natural language) using self-supervision.
- General Purpose: Unlike "narrow AI" (e.g., just flood detection), FMs learn general representations of the Earth.
- The "Backbone" serves as a universal feature extractor upon which specialized "heads" are built for downstream tasks (e.g, flood detection, carbon monitoring, detecting methane emissions, etc).



GFMs are trained globally

Typical workflow of spatio-temporal models





- High label dependency
- Brittle to domain shift

- Self supervised learning
- Multi-modal fusion
- More robust to OOD?

The Data Challenge

01

Earth Observation (EO) and Climate data has reached the petabyte scale. Labeling this data is prohibitively expensive and requires deep domain expertise.

02

The Solution: GeoFMs leverage unlabeled data to learn patterns without human supervision, directly addressing label scarcity.

1.6TB

Daily Data from Sentinel-2

Models and Scale

Scale and Data →

Classical ML, Simple Neural Networks

Task-specific handcrafted feature representations

Deep Neural Networks

Learnt feature representations

Foundation Models

Generalizable & adaptable learnt representations

← **Interpretability**

Origin of Foundation Models

2018: NLP Beginnings

BERT and GPT 1 introduce self-supervised learning to learn language structure.

2020: CV Adaptation

Vision Transformers (ViT) and Masked Autoencoders apply SSL to images.

2022: Geo Gap

Direct application of CV models fails, need for "native" geospatial architectures arises.

2022+: GFM Era

Specialized models like Prithvi, SatMAE arrive.

Over 30+ Foundation Models

- TerraMind (IBM, ESA, JSC)
- Prithvi (IBM, NASA)
- AlphaEarth (Google)
- CopernicusFM (TU Munich)
- Tessera (Cambridge)
- OceanRep (Alfred Wegener Institute)
- THOR (NR, ESA)

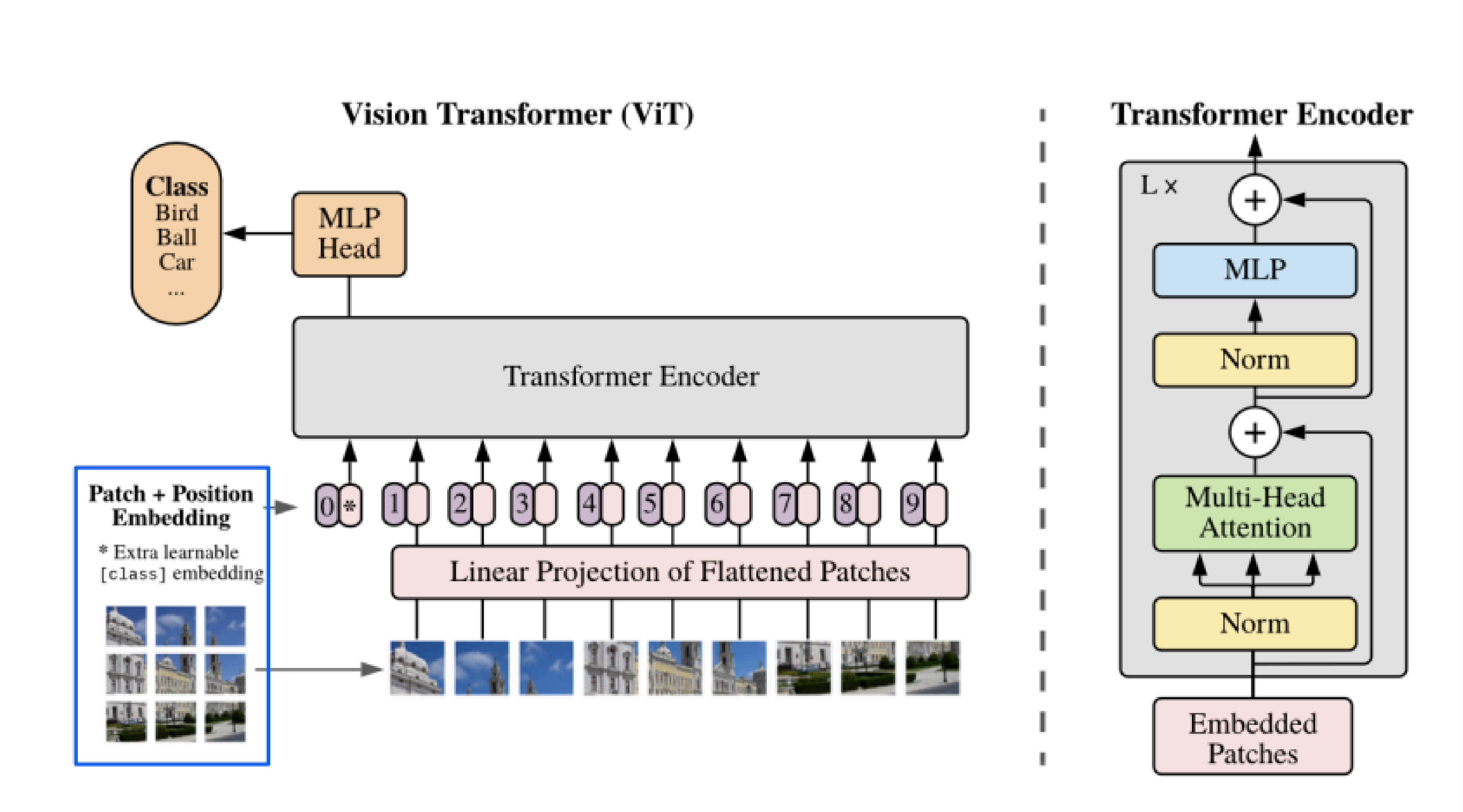
https://github.com/Jack-bo1220/Awesome-Remote-Sensing-Foundation-Models

Vision Transformers

Patching: The satellite image is broken into small square patches (e.g., 16x16 pixels).

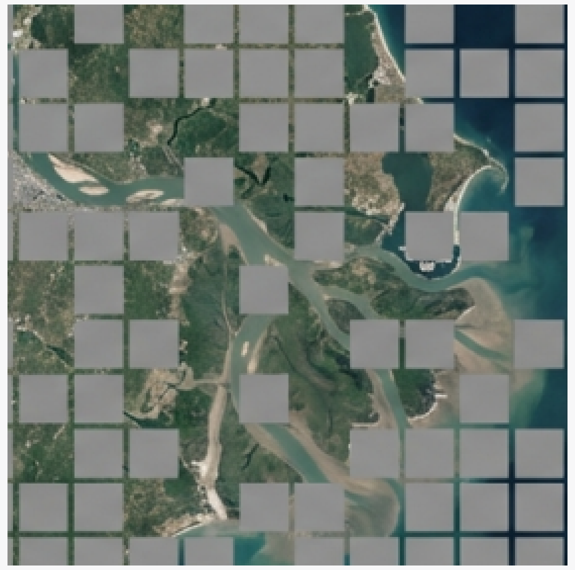
Embedding: Each patch is flattened and linearly projected into an embedding vector.

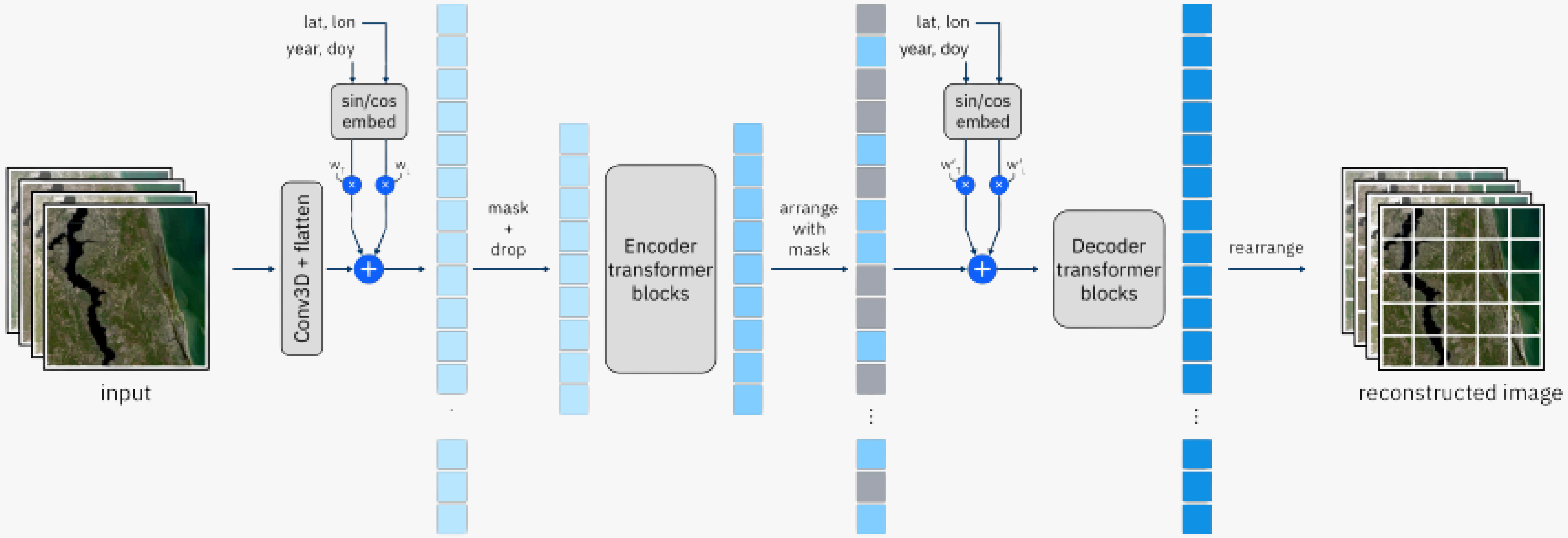
Attention: Self-attention mechanisms allow the model to learn global context.



Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

Masked Image Modeling (MIM)





Characteristics of GFMs

Multimodal

Ingests Optical (RGB), SAR (Radar), DEM (Elevation), and Meteorological data simultaneously.

Multi-scale

Functions across resolutions, from 30m Landsat archives to sub-meter commercial imagery.

Spatiotemporal

Inherently understands time (seasonality) and location (coordinates), not just static pixels.

Downstream applications

Segmentation

Pixel-level classification for land cover mapping, crop type identification, and water detection.

Regression

Predicting continuous values such as Biomass estimation and Canopy height.

Temporal Analysis

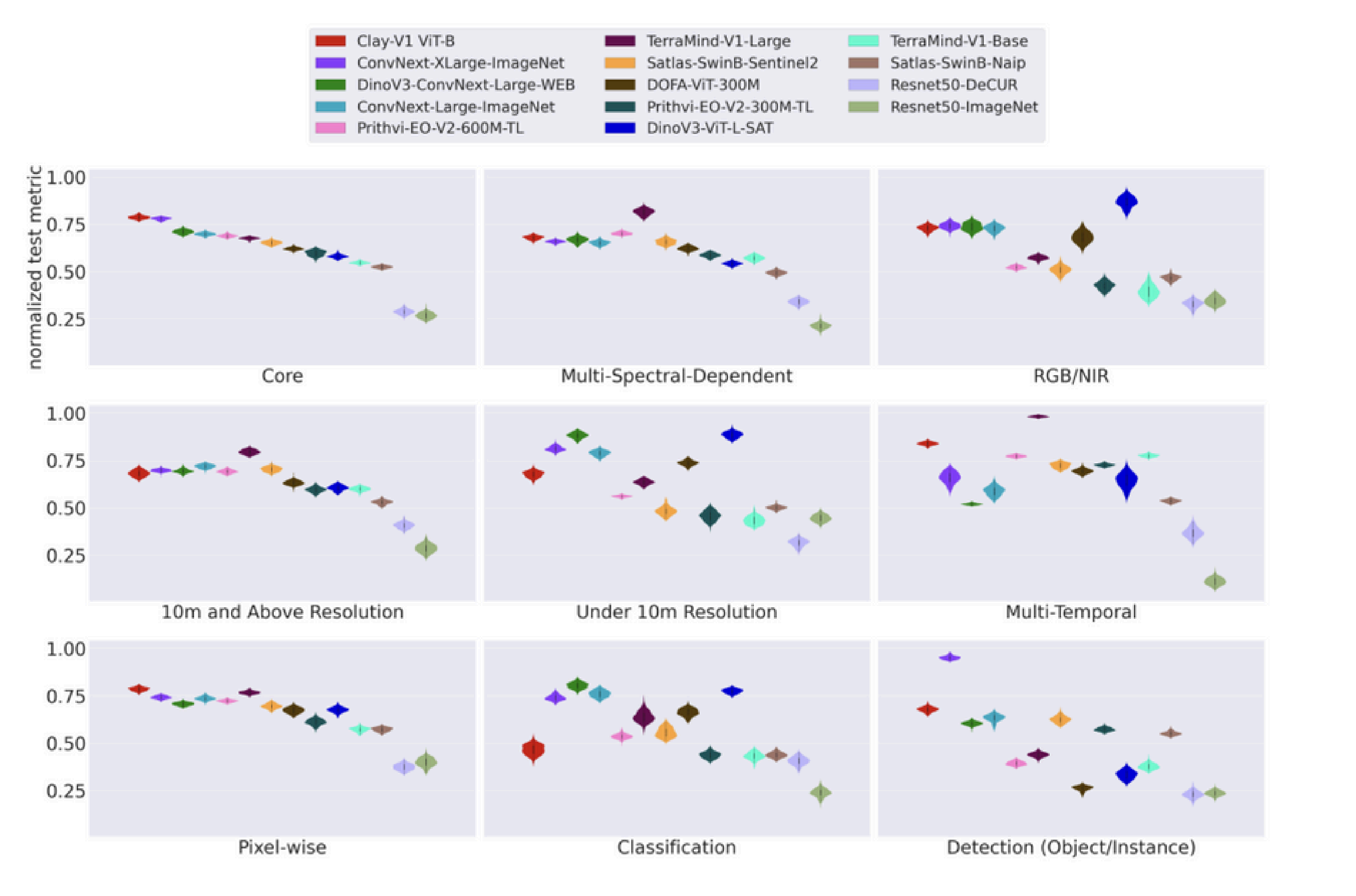
Change detection for monitoring deforestation, urbanization, and flood progression.

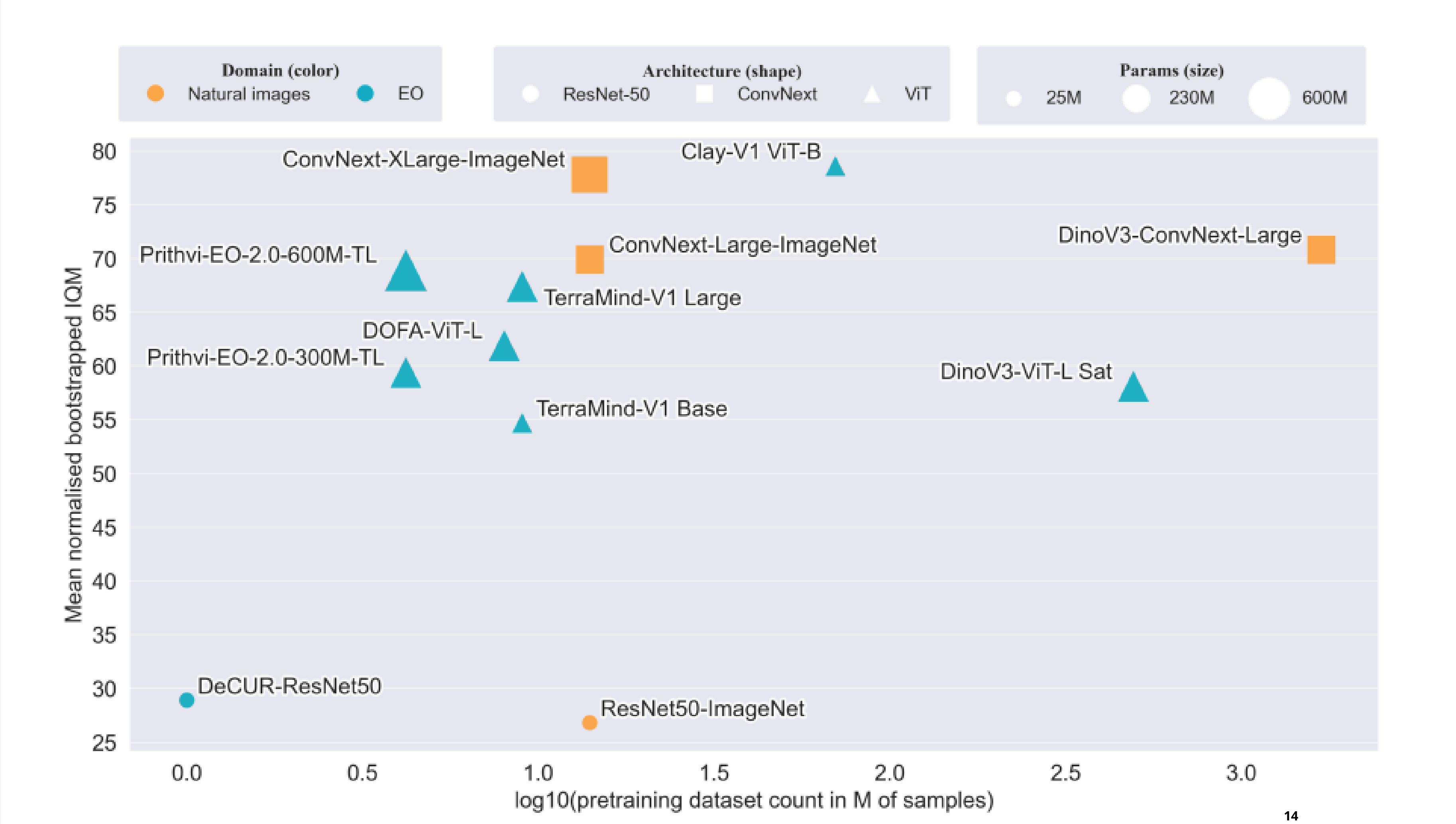
Community Benchmarks

GeoBench 1 and 2

PANGAEA

REOBench





Pre-training and Fine-tuning

01

Pre-training

- Large/huge unlabeled dataset
- Goal: Learn general representations
- Requires 10s to 100s of GPUs

02

Frozen encoder/Feature extractor

- encoder weights (pre-trained) are frozen
- only decoder is trained for the specific task
- Requires a single GPU

03

Fine-tuning

- Smaller labeled dataset (task-specific)
- Goal: learn specific task.
- Can usually be done with a single GPU

Resources

- TerraTorch https://terrastackai.github.io/terratorch/stable/
- Ready to use AlphaEarth embeddings https://developers.google.com/earth-engine/datasets/catalog/GOOGLE_SATELLITE_EMBEDDING_V1_ANNUAL
- Tessera embeddings https://github.com/ucam-eo/tessera
- https://github.com/VMarsocci/pangaea-bench
- https://github.com/ServiceNow/geo-bench
- https://www.fast-eo.eu/use-cases

2nd ESA-NASA Workshop on AI Foundation Model https://nikal.eventsair.com/2nd-esa-nasa-workshop-on-ai-foundation-model-for-earth-observation-eo/hands-on-session-submission

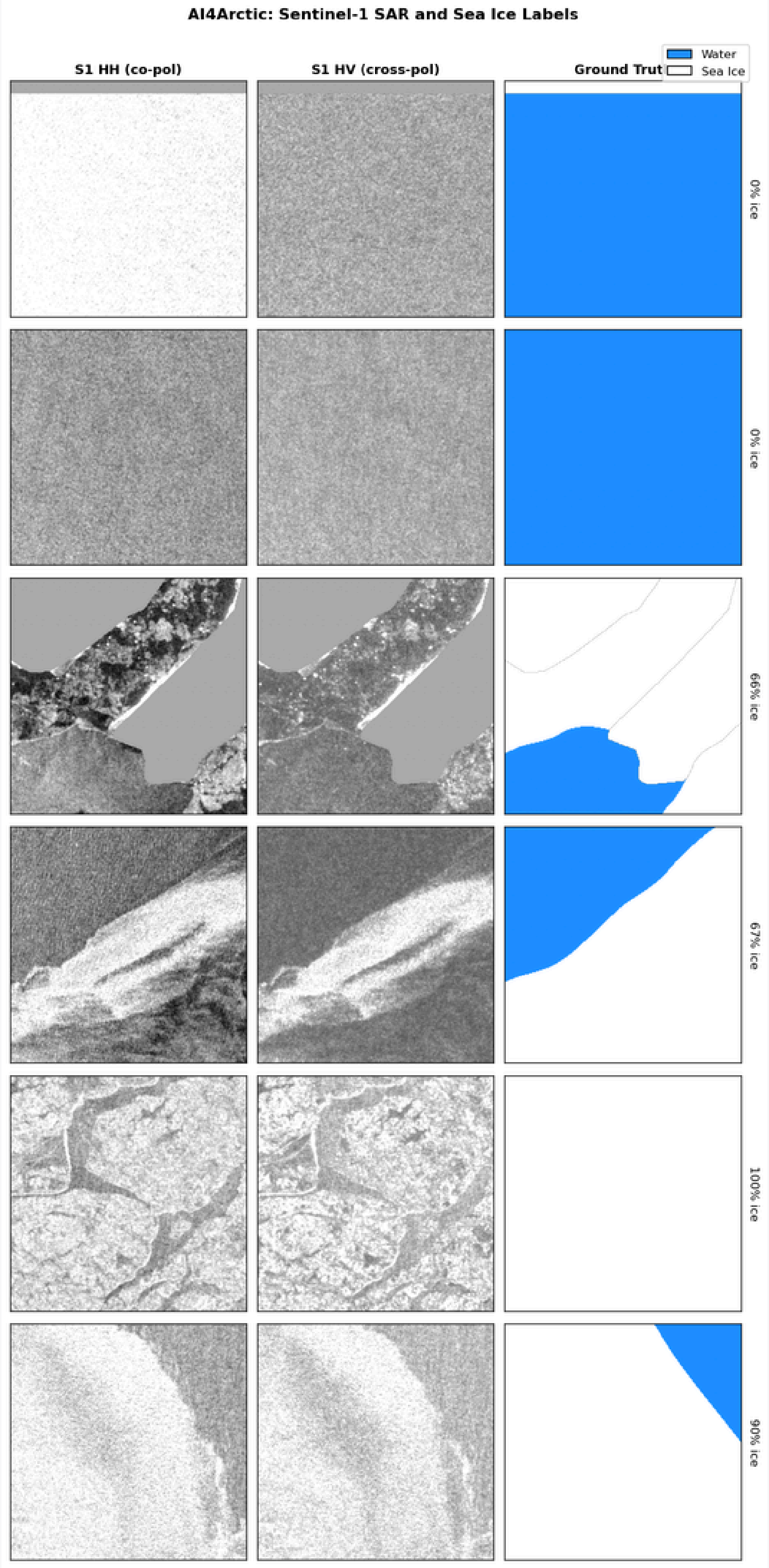
Sea Ice Segmentation with TerraMind

Goal: Detect sea Ice in Sentinel-1 images

Data: The AI4Arctic benchmark dataset contains 512 Sentinel-1 EW mode SAR scenes covering Arctic sea ice regions (2018-2021).

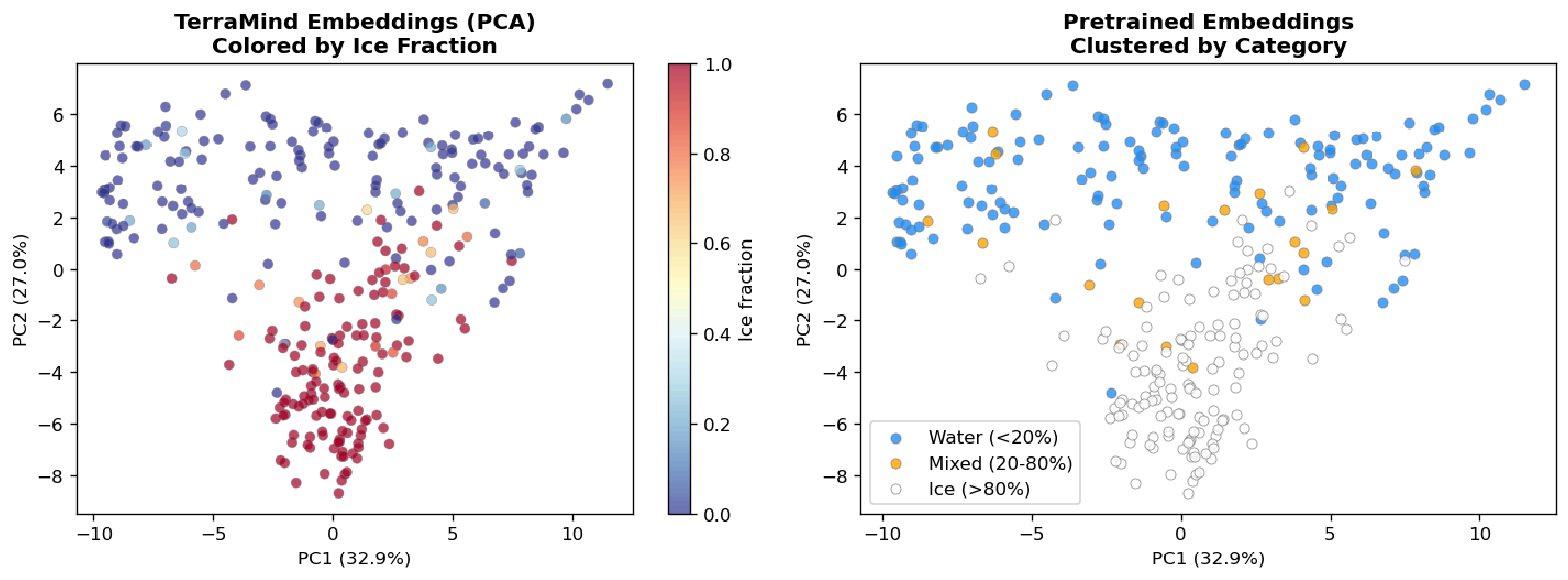
Each scene includes:

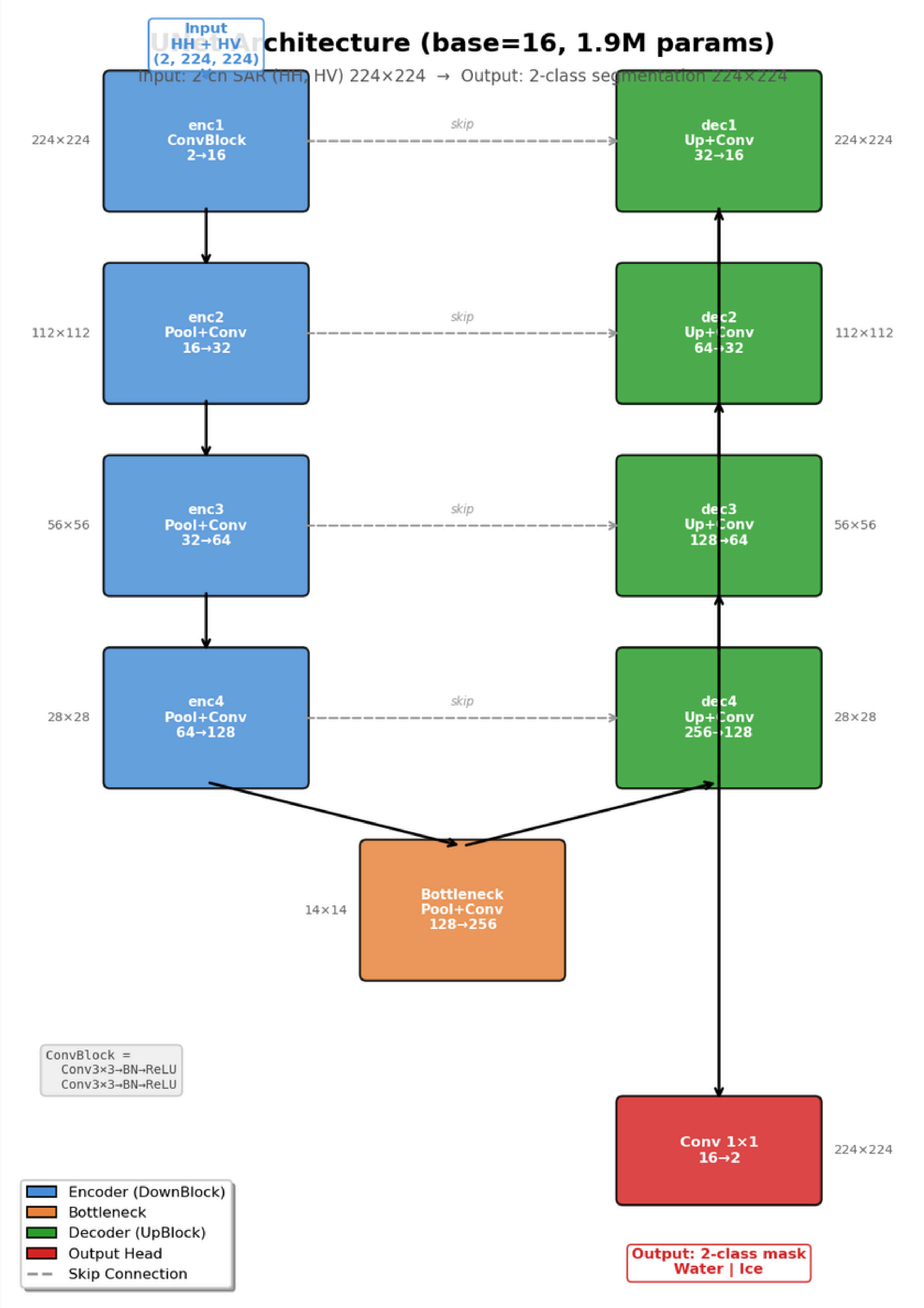
- HH (co-polarization) and HV (cross-polarization) backscatter
- Sea Ice Concentration (SIC) labels as class indices 0-10 (each ~10% concentration bin)
- We preprocessed these into 224×224 patches .



Step 2: Extract embeddings from Terramind

1. **PCA:** do embeddings naturally separate ice from water?
2. **kNN probing:** how well can a simple nearest-neighbor classifier segment ice using raw embeddings?





Step 3: Train a U-Net Baseline

```
python -m sea_ice.evaluation.train --model unet --gpu 0
```

UNet (base_channels=16, 1.9M parameters) for 50 epochs

Step 4: Train TerraMind (Probing: frozen backbone)

```
python -m sea_ice.evaluation.train --model terramind --mode probing --gpu 0
```

Freeze the TerraMind backbone and train only a segmentation head (3-layer conv head) (30 epochs).

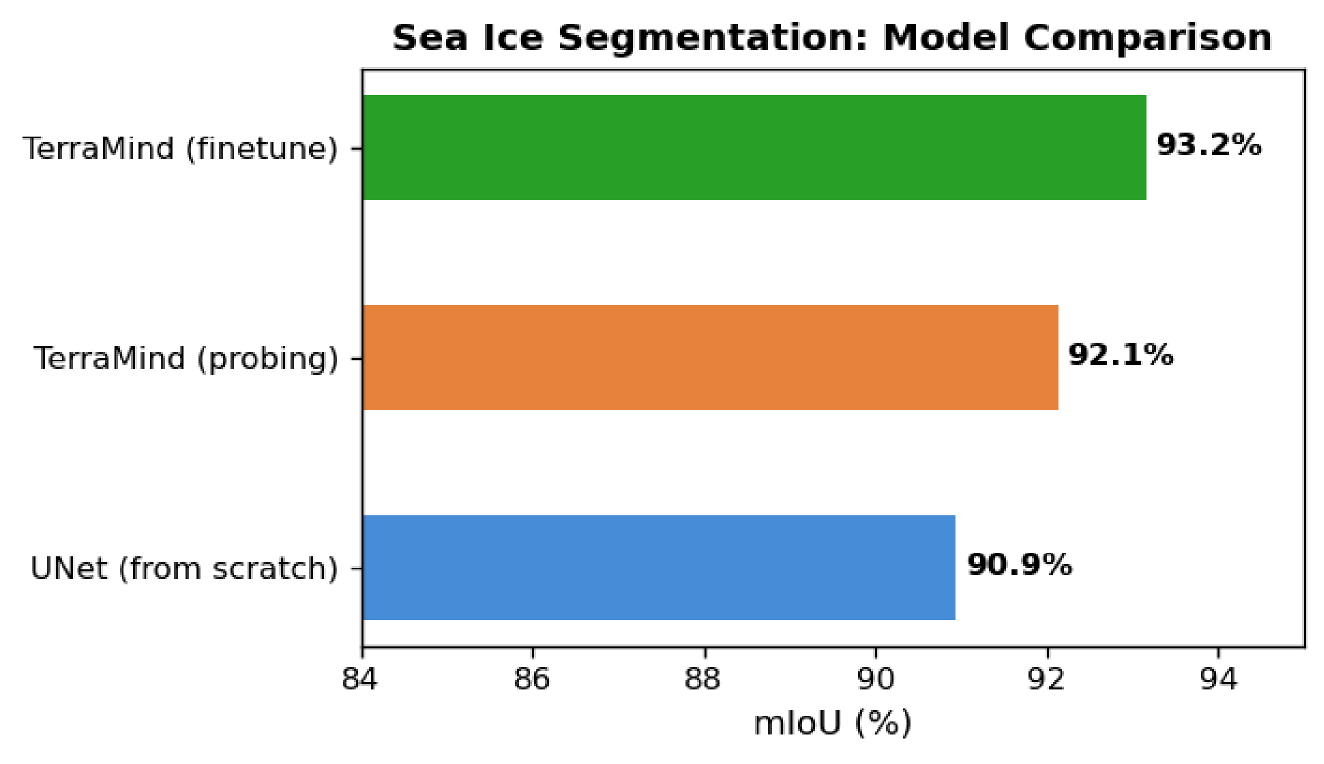
Step 5: Train TerraMind (Fine-tuning)

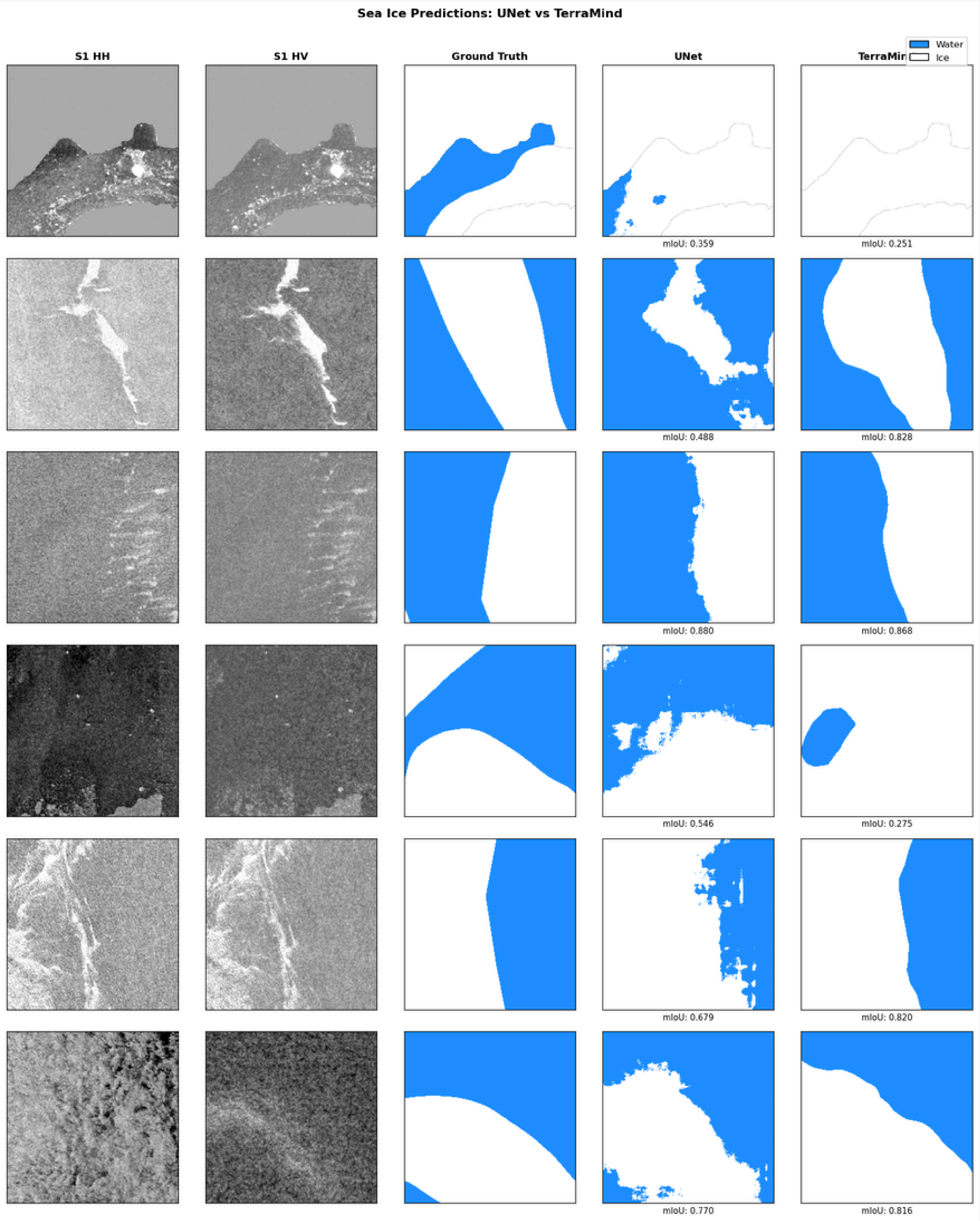
```
python -m sea_ice.evaluation.train --model terramind --mode probing --gpu 0
```

Fine-tune the full TerraMind model with a lower learning rate (1e-5, 50 epochs).

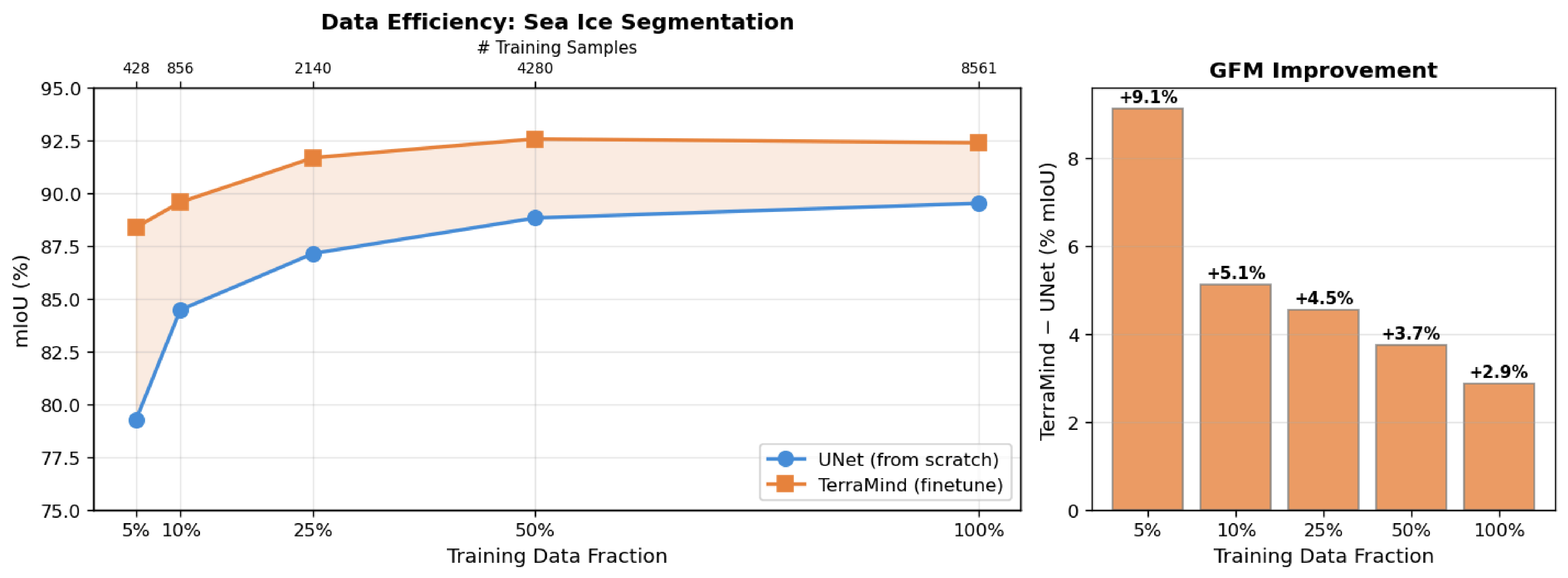
Results

Model	Mode	mIoU (%)	Params
UNet	From scratch	89.5	1.9M
TerraMind	Probing (frozen)	91.0	2.4M trainable
TerraMind	Fine-tune	**92.4**	87.7M





What if we have much less data?




```
python -m sea_ice.evaluation.data_efficiency --gpu 0 --fractions 0.05,0.1,0.25,0.5,1.0
```

Training Data	UNet mIoU	TerraMind mIoU	Gap
5% (428 patches)	79.3%	88.4%	+9.1%
10%	84.5%	89.6%	+5.1%
25%	87.2%	91.7%	+4.5%
50%	88.8%	92.6%	+3.7%
100% (8,561 patches)	89.5%	92.4%	+2.9%

Try GFMs for your own classification, segmentation, regression, forecasting applications.

Propose challenging tasks for community benchmarks.