Image from NASA:
https://commons.wikimedia.org/wiki/File:Earth_Western_Hemisphere_transparent_background.png#filelinks

Geospatial Deep Learning

Foundation Models

WV VIEW

AmericaView
Empowering Earth Observation Education
AmericaView.org - Est. 2003

The primary focus of this short module is foundation models. We begin with a review of some key concepts of learning from unlabeled data, as this is a key component of foundation models.

Module Topics

1. Unsupervised vs. semi-supervised vs. self-supervised vs. supervised learning
2. Means to incorporate unlabeled data
3. Characteristics of foundation models
4. Segment Anything Model (SAM)
5. AlphaEarth

2

These are the topics covered in this module.

Terminology

- Unsupervised
  - Learn from unlabeled data
  - Identify clusters
  - Clusters may or may not map well to classes of interest
- Semi-Supervised
  - Train on large set of unlabeled data and a smaller set of labeled data
  - Losses
    - Classification accuracy
    - Label-consistency or smoothness
- Self-Supervised
  - Two-stage
  - Train on large set of unlabeled data
  - Fine-tune on labeled dataset
- Supervised
  - Only use labeled data

This slide introduces the key terminology for different learning paradigms. Unsupervised learning uses no labels and focuses on finding structure such as clusters in data. Semi-supervised learning uses a large unlabeled dataset alongside a smaller labeled set, with losses for both classification accuracy and label consistency. Self-supervised learning is a two-stage approach: pre-train on unlabeled data using automatically generated supervision, then fine-tune on a labeled dataset. Supervised learning relies entirely on labeled data and is the most common approach covered in prior modules.

Assumptions

- Smoothness Assumption = if two samples are close together in the feature space, they should have the same label
- Low-Density Assumption = class boundaries should occur in low-density regions of the feature space (where there are not many samples present)
- Manifold Assumption = the input feature space is composed of multiple lower-dimensional manifolds on which all data points lie and data points lying on the same manifold have the same label

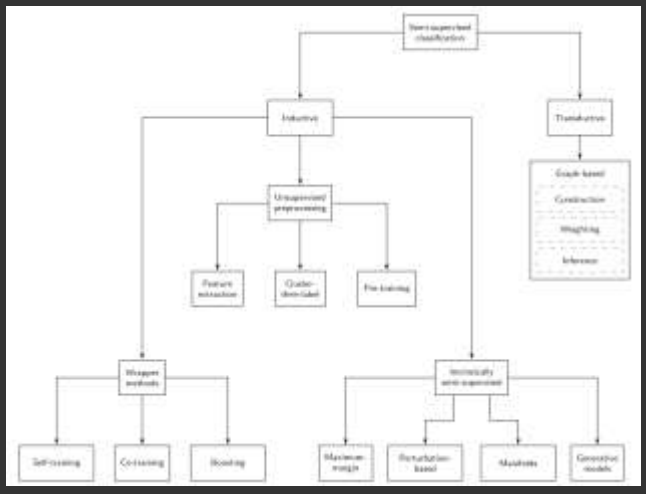
Van Engelen, J.E. and Hoos, H.H., 2020. A survey on semi-supervised learning. *Machine learning*, *109*(2), pp.373-440.
https://link.springer.com/article/10.1007/s10994-019-05855-6

Learning from unlabeled data rests on three core assumptions about the data. The Smoothness Assumption holds that nearby samples in feature space should share the same label. The Low-Density Assumption states that decision boundaries should pass through areas with few data points, avoiding densely populated regions. The Manifold Assumption posits that high-dimensional data lies on lower-dimensional structures (manifolds), and samples on the same manifold share labels. These assumptions justify using the structure of unlabeled data to inform classification boundaries.

Learning Types

- Inductive = find a classification model
- Transductive = obtain predictions for unlabeled data points

Van Engelen, J.E. and Hoos, H.H., 2020. A survey on semi-supervised learning. *Machine learning*, *109*(2), pp.373-440.



Semi-supervised learning can be approached in two ways. Inductive methods aim to learn a general classification model that can predict labels for any new input, including unseen data. Transductive methods focus only on predicting labels for the specific unlabeled data points available during training, without generalizing to a broader model. Most deep learning approaches are inductive, but transductive methods can be effective when the unlabeled test set is fixed and known in advance.

Wrapper Methods

1. Train supervised classifier on labeled data
2. Use trained classifier to predict to unlabeled data
3. Re-train on pseudo-labelled data + labeled data

- Self-Training = supervised classifier iteratively trained on both labeled and pseudo-labeled data from prior iterations
- Co-Training = two or more supervised classifiers are iteratively trained on the labeled data, adding their most confident predictions to the labeled data set of the other supervised classifiers in each iteration
- Bagging = train separate classifiers on subset of training data; aggregate results
- Boosting = iterative; apply higher weight to samples that were misclassified by prior classifier

Wrapper methods treat the supervised classifier as a black box and use it iteratively with unlabeled data. Self-training iteratively trains the classifier, uses it to assign pseudo-labels to unlabeled examples, then retrains on both labeled and pseudo-labeled data. Co-training uses two or more classifiers that each pseudo-label data for the other, effective when different feature views of the data are available. Bagging and boosting are ensemble techniques: bagging trains on random subsets and aggregates predictions, while boosting sequentially reweights misclassified samples.

Unsupervised Pre-Processing

1. Extract useful features from unlabeled data, pre-cluster data, or parameterize supervised learning procedure
2. Supervised algorithm only uses labeled data

- Feature Extraction = transform input data into an improved or computationally more efficient feature space
  - Principal Component Analysis (PCA) and Kernel Principal Component Analysis (kPCA)
  - Autoencoders and Denoising Autoencoders
- Cluster-then-Label = apply unsupervised or semi-supervised clustering algorithm to all available data; use clusters to guide supervised classification
  - Self-Organizing Maps
- Pre-Training = unlabeled data is used to guide the decision boundary towards potentially interesting regions before applying supervised training
  - Deep Belief Networks
  - Stacked Autoencoders

Unsupervised pre-processing leverages unlabeled data to improve the feature representation before supervised learning begins. Feature extraction methods like PCA, kernel PCA, autoencoders, and denoising autoencoders transform inputs into more compact or informative representations. Cluster-then-label methods apply clustering algorithms to all data and use the resulting cluster assignments to guide supervised learning. Pre-training methods like Deep Belief Networks and Stacked Autoencoders use unlabeled data to initialize model weights in a meaningful region of parameter space before supervised fine-tuning.

Intrinsically Semi-Supervised

- Maximum-Margin = augment maximum-margin-based classifiers, such as support vector machines (SVM), with unlabeled data to inform areas of low density
- Perturbation-Based = add noise to improve generalization
- Manifolds = considers manifolds in data perturbations
- Generative Models = learn underlying structure of data by attempting to understand its underlying characteristics and distribution
  - Generative Adversarial Networks (GANs)
  - Variational Autoencoders (VAEs)

Intrinsically semi-supervised methods integrate labeled and unlabeled data into the learning algorithm itself rather than using them in separate stages. Maximum-margin methods extend SVMs by using unlabeled data to push decision boundaries into low-density regions. Perturbation-based methods add noise to inputs and enforce consistent predictions, improving robustness. Manifold-based methods incorporate the structure of data manifolds into the regularization. Generative models like GANs and VAEs learn the data distribution from unlabeled data and can be combined with classification objectives.

Diffusion

- Add Gaussian noise to images
- Learn to reconstruct images
- Noise is added sequentially during forward diffusion process
- Reverse diffusion process is learned to remove noise or restore original data in a stepwise fashion
- Use modified UNet model

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, S., 2015, June. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256-2265). PMLR.

Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, pp.6840-6851.

Diffusion models are generative models that learn to reconstruct data by reversing a gradual noise-adding process. In the forward diffusion process, Gaussian noise is added to an image incrementally over many steps until the image becomes pure noise. The model then learns the reverse process: given a noisy image, predict and remove the noise step-by-step. A modified UNet architecture is typically used for this denoising task. At inference time, starting from random noise, the reverse diffusion process generates realistic new images. Diffusion models underpin image generation systems like Stable Diffusion and DALL-E 2.

Unsupervised Learning Tasks for Image-Like Data

1. Rotation angle
2. Gap filling
3. Noise removal
4. Lat/long prediction
5. Seasonality prediction

Self-supervised learning for image-like data (including remote sensing imagery) creates supervision signals from the data itself without human labels. Common pretext tasks include predicting rotation angle, filling in masked or missing regions (gap filling), removing artificially added noise, predicting geographic coordinates (lat/long) from image content, and predicting seasonal or temporal characteristics. These tasks force the model to learn meaningful representations of the data that can then be transferred to downstream tasks with limited labeled data.

Foundation Models

- Large models with many trainable parameters
- Rely on semi-supervised learning
- Transformer architectures commonly used
- Trained on a large volume of variable data for generalizability
- Can be fine-tuned for specific tasks
- Examples: ChatGPT-3, ChatGPT-4, BERT, DALL-E, Stable Diffusion

https://www.earthdata.nasa.gov/news/impact-ibm-hls-foundation-model

Foundation models are large-scale models pre-trained on massive and diverse datasets that can be adapted to a wide range of downstream tasks. They typically use Transformer architectures and rely on semi-supervised or self-supervised learning to train on unlabeled data at scale. After pre-training, they can be fine-tuned for specific applications with relatively small labeled datasets. Examples include GPT-3 and GPT-4 for text, BERT for language understanding, and DALL-E and Stable Diffusion for image generation. In remote sensing, foundation models like Prithvi and SatMAE have been developed.

What is a Foundation Model?

1. General-purpose model that can be fine-tuned for specific tasks
2. Pre-trained on large and diverse datasets
3. Learn representations that can support a wide variety of tasks
4. Adaptable to downstream tasks
5. Generally large with a large number of trainable parameters
6. Trained without labels using self-supervised or weakly-supervised methods
7. Emergent capabilities

Foundation models are distinguished from task-specific models by several key characteristics. They are pre-trained on large and diverse datasets, enabling them to learn broadly useful representations. They are designed to be adaptable to many downstream tasks through fine-tuning. They are generally very large, with billions of trainable parameters. They are commonly trained using self-supervised or weakly-supervised methods that do not require manual labels. Perhaps most significantly, foundation models exhibit emergent capabilities: behaviors that were not explicitly trained for but arise from the scale and diversity of training.

Segment Anything

- Computer vision foundation model
- Generated by Meta
- For general-purpose image segmentation
- Currently, versions 1, 2, and 3 available

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y. and Dollár, P., 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4015-4026).

13

We now provide a few examples of foundation models that have been found to be useful in remote sensing and geospatial science.

Meta's Segment Anything Model (SAM) series, developed by Meta AI, represents a progression of increasingly capable segmentation foundation models, with each generation expanding what kinds of inputs the model can understand and what kinds of data it can operate on.

SAM 1 (released April 2023) established the foundation for promptable, zero-shot image segmentation. Trained on the large-scale SA-1B dataset, SAM 1 supports spatial prompts including points, bounding boxes, and masks, enabling strong zero-shot interactive segmentation without task-specific fine-tuning. Its architecture consists of an image encoder, a prompt encoder, and a mask decoder, and it can automatically segment all objects in an image or isolate specific regions based on a user-provided prompt. Crucially, SAM 1 operates only on static images and has no understanding of natural language or temporal context.

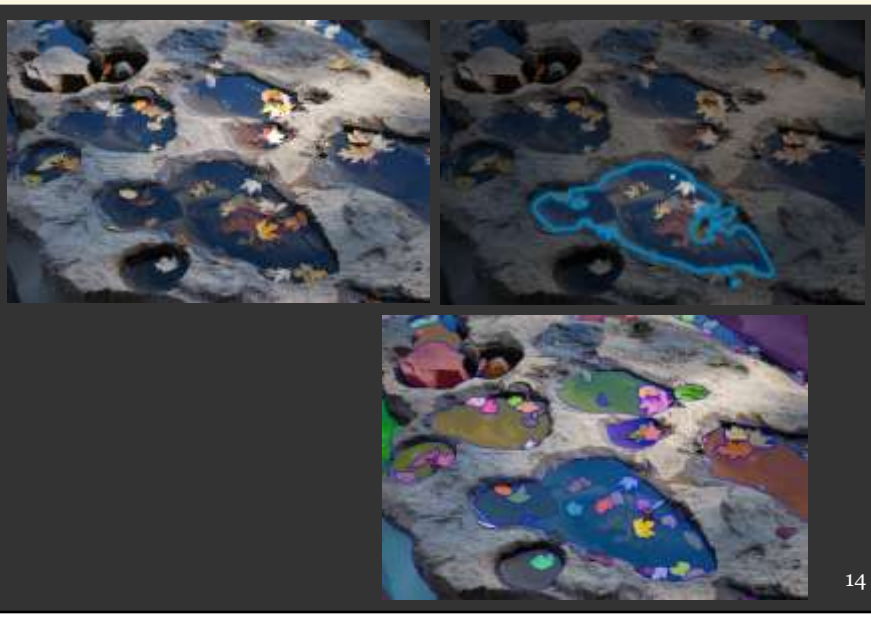
SAM 2 (released mid-2024) adds support for video. SAM 2 introduced a unified image and video segmentation capability, with a streaming memory module that tracks objects across frames. This temporal key-value memory supports long-sequence video reasoning and maintains object identity across

frames, combining ViT-based visual tokens with a memory retrieval pathway to propagate masks over time.

SAM 3 (released November 2025) adds Promptable Concept Segmentation (PCS), where a short text phrase or example image tells the model to segment every instance of a concept in an image or video.

Segment Anything

- Image encoder
- Image embeddings
- Prompt Encoder
- Mask Encoder



https://segment-anything.com/
https://samgeo.gishub.org/

Its architecture has three main components: an Image Encoder that converts input images into dense image embeddings using a Vision Transformer; a Prompt Encoder that converts user-provided prompts (points, boxes, masks, or text) into prompt embeddings; and a Mask Decoder that combines image and prompt embeddings to generate segmentation masks.

Segment Anything

Version	Release Year	Main idea	Inputs/prompts	Main output	Biggest advance
SAM 1	April 2023	Promptable segmentation for images	Points, boxes, masks	Masks for prompted regions or objects	General-purpose, class-agnostic image segmentation
SAM 2	July 2024	Promptable segmentation for images and video	Points, boxes, masks across frames	Masks that can persist through video	Adds temporal memory and video object segmentation
SAM 3	November 2025	Promptable concept segmentation for images and video	Text, exemplars, points, boxes, masks	Detects, segments, and tracks all matching concept instances	Adds open-vocabulary text/exemplar-based segmentation

Assumption	Description
Image Encoder	Convert input images into dense image embeddings
Prompt Encoder	Convert text, points, boxes, or mask into prompts
Mask Decoder	Combines images and prompt information to generate segmentation masks

This slide details the three versions of Segment Anything. SAM 1 (April 2023) introduced promptable segmentation for static images using points, boxes, or masks as inputs. SAM 2 (July 2024) extended this to video, adding temporal memory so masks persist across frames for video object segmentation. SAM 3 (November 2025) added open-vocabulary segmentation using text and exemplar inputs, enabling the model to detect, segment, and track all instances of a concept described in natural language. As noted on the prior slide, the architecture consists of three components: an Image Encoder, a Prompt Encoder, and a Mask Decoder.

Fine-Tuning SAM

- Zero-Shot = no new labels (just provide new data and obtain the segmentation)
- Few-Shot = Given small set of labels

Method	What is trained?
LoRA	Low-rank updates inside attention/projection layers
Adapters	Small trainable layers inserted into the encoder or decoder
Prompt Tuning	Learned prompt embeddings or prompt generator
Decoder-Only Fine-Tuning	Mask Decoder
Input Adapter	Small network that maps non-standard input bands into SAM-compatible inputs

Low-Rank Adaption (LoRA) = Keep the original model weights frozen, then add small trainable low-rank matrices that learn task-specific adjustments

16

Zero-shot learning and few-shot learning both address the same fundamental challenge: how can a model perform a task it was never explicitly trained on? They differ, however, in how much task-specific information is available at inference time.

In zero-shot learning, the model receives no examples of the target task whatsoever; it must rely entirely on knowledge and representations built up during pretraining, along with a natural language description or instruction explaining what is expected.

Few-shot learning provides the model with a small number of input-output examples before asking it to perform the task on a new input. These examples are included directly in the prompt and serve as an implicit guide to the format, style, and logic of the expected response.

SAM can be fine-tuned for domain-specific tasks using several lightweight adaptation strategies. LoRA (Low-Rank Adaptation) keeps the original weights frozen and adds small trainable low-rank matrices inside attention layers, enabling efficient adaptation with few parameters. Adapters insert small trainable modules into the encoder or decoder. Prompt Tuning learns a set of prompt embeddings rather than modifying model weights. Decoder-Only Fine-

Tuning trains only the mask decoder. Input Adapters handle non-standard inputs such as multi-spectral or hyperspectral remote sensing bands that differ from SAM's expected RGB input.

SAM for Geospatial Data

- In Python SamGeo — https://samgeo.gishub.org/
- InGIS (qgis-sam-geo-plugin) — https://github.com/opengeos/qgis-samgeo-plugin
- In ArcGIS — https://www.arcgis.com/home/item.html?id=9b67b441f29f4ce6810979f5f0667ebe

SAM has been integrated into several geospatial tools. In Python, the samgeo library provides a convenient interface for applying SAM to geospatial raster data, including satellite imagery. A QGIS plugin (qgis-samgeo-plugin) allows GIS users to run SAM directly within the QGIS desktop environment for interactive segmentation. ArcGIS also offers SAM integration through a dedicated item in the ArcGIS Online marketplace, enabling Esri users to leverage SAM for feature extraction and mapping tasks.

AlphaEarth

- Developed by Goole DeepMind
- General-purpose Earth observation embeddings
- 64 embeddings
- 10 m spatial resolution
- Derived from
- Global extent
- Annual update (2017 to 2014)
- Input data = Sentinel-2 L1C, Sentinel-1 GRD, Landsat 8/9 Tier 1 ToA, and others (GEDI, GRACE, GLO-30 DEM, NLCD)

Brown, C.F., Kazmierski, M.R., Pasquarella, V.J., Rucklidge, W.J., Samsikova, M., Zhang, C., Shelhamer, E., Lahera, E., Wiles, O., Ilyushchenko, S. and Gorelick, N., 2025. Alphaearth foundations: An embedding field model for accurate and efficient global mapping from sparse label data. *arXiv preprint arXiv:2507.22291*.

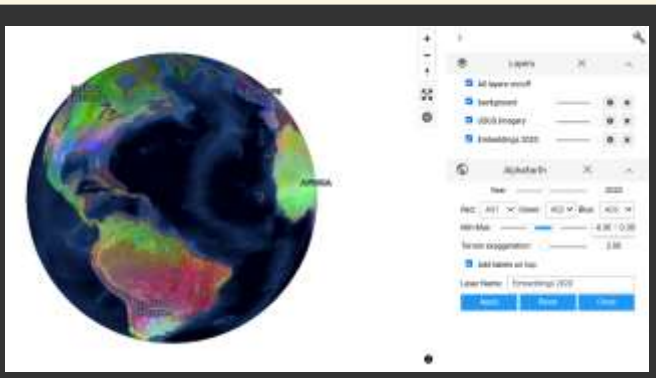
18

AlphaEarth Foundations is a geospatial foundation model developed by Google DeepMind, announced in mid-2025. It is designed to function like a virtual satellite, characterizing the planet's terrestrial land and coastal waters by integrating large amounts of Earth observation data into a unified digital representation, or embedding, that computer systems can process.

The model's core approach centers on creating compact, information-rich summaries of the Earth's surface. It divides the planet into 10-by-10 m cells, each containing a compact summary of what's there, updated annually, automatically fusing and standardizing data to create a consistent view of the planet. The data sources ingested include optical and thermal imagery from Sentinel-2 and Landsat satellites, radar data that can see through clouds, 3D measurements of surface properties, global elevation models, climate information, gravity fields, and descriptive text. Each location is represented as a 64-dimensional embedding that distills complex optical, radar, and climate inputs into an analysis-ready format.

AlphaEarth Via Google Earth Engine

- Made available via Google Earth Engine



https://developers.google.com/earth-engine/tutorials/community/satellite-embedding-01-introduction

https://developers.google.com/earth-engine/datasets/catalog/GOOGLE_SATELLITE_EMBEDDING_V1_ANNUAL

https://opengeoai.org/examples/AlphaEarth/

19

The AlphaEarth embedding are available via Google Earth Engine.

This is the end of this lecture module.

Please return to the West Virginia View Webpage for additional content.

20

Thanks! Hope you found this useful.