

Developing Reliable Machine Learning Methods for Earth Observation and Landscape Change

Ping-Chun Lin

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2026

Reading Committee:

Akshay Mehra, Chair

Marine Denolle

Hayato Ushijima-Mwesigwa

Program Authorized to Offer Degree:

Earth and Space Sciences

©Copyright 2026

Ping-Chun Lin

University of Washington

Abstract

Developing Reliable Machine Learning Methods for Earth Observation and Landscape Change

Ping-Chun Lin

Chair of the Supervisory Committee:

Akshay Mehra

Department of Earth and Space Sciences

Machine learning (ML) and artificial intelligence (AI) models have become popular research tools for geospatial analysis, but the standard AI/ML metrics that guide model development do not necessarily show whether an output supports the scientific task. The difference between evaluation benchmarks and downstream utility creates a “reliability gap” between developers and users of geospatial AI workflows. This dissertation develops methods that incorporate “reliability” into geospatial AI workflows in multiple stages, so that model outputs remain aligned with the downstream tasks and decisions. The central idea is that by incorporating domain knowledge, standardized expert feedback loops, and task-relevant training strategies from the user perspective, geospatial AI workflows can be made more reliable for users.

This central idea is supported by three chapters that incorporate guidance at different stages of the workflow across multiple Earth observation applications. In the first chapter, I incorporate statistical priors into generative AI models for satellite image quality enhancement. I introduced a curriculum-guided diffusion model, conditioned on a classical speckle filter, the Lee filter, that removes speckle noise from synthetic aperture radar (SAR) imagery while preserving the structures that downstream object detection requires, with ship detection improvements on SAR imagery

outside the training data. In the second chapter, I standardize expert feedback protocols for transparency and downstream utility. I demonstrated a framework for expert-in-the-loop workflows that converts object detections into counts, areas, and calibrated uncertainties for lined farm ponds in Maharashtra, India, with transparency and replicability that support decision-making. The third chapter incorporates spatiotemporal and environmental context into foundation model evaluation. I test the consistency of geospatial foundation model embeddings for predicting major soil elements across the conterminous United States, which shows that performance depends on sampling density, embedding-space proximity, and geological context, and that adding geology and soil covariates improves prediction in regions where field measurements are scarce and out of the training data distribution.

These three chapters show that geospatial AI workflow outputs become scientifically useful when downstream tasks, expert review, and environmental context guide both workflow design and evaluation. Reliability, when designed into multiple stages of the workflow, gives users justified confidence in the resulting measurements.

Contents

Abstract	iii
List of Figures	viii
List of Tables	x
1 Acknowledgments	xi
2 Introduction	1
2.1 Background	1
2.2 The Gap in Geospatial AI Modeling	2
2.3 Designing Reliability into Geospatial AI Workflows	2
2.4 Expert Guidance in the Geospatial AI Workflow	4
3 Chapter 1: Expert Guided Diffusion for Synthetic Aperture Radar Despeckling	6
3.1 Relevance	6
3.2 Status	6
3.3 Chapter Abstract	6
3.4 Introduction	7
3.4.1 Speckle Noise Model	8
3.4.2 Diffusion Models	9
3.5 Methodology	11
3.5.1 SAR-DDPM and the Loss Equation	11
3.5.2 Training Framework	12
3.6 Experiments	13
3.6.1 Training Data	13
3.6.2 Experiments	14
3.7 Results and Discussion	15
3.7.1 Results	15
3.7.2 Discussion	16
3.8 Conclusion	18

4	Chapter 2: Incorporating expert inputs to produce calibrated outputs: Building confidence in applications of machine learning to remote sensing	19
4.1	Relevance	19
4.2	Status	19
4.3	Chapter Abstract	19
4.4	Introduction	20
4.5	Methodology	22
4.5.1	Overall Framework	22
4.5.2	Preprocessing	22
4.5.3	Training	24
4.5.4	Post-processing	28
4.6	Case Study	28
4.6.1	Lined Farm Ponds in Maharashtra, India	28
4.6.2	Data	29
4.6.3	Workflow application	30
4.6.4	Uncertainty Calibration	31
4.7	Results and Discussion	32
4.7.1	Results	32
4.7.2	Discussion	35
4.8	Conclusion	36
4.9	Open Research	37
5	Chapter 3: Less is More? Evaluating Geospatial Foundation Models for Soil Geochemistry Prediction Under Data Scarcity and Geographic Shift	40
5.1	Relevance	40
5.2	Status	40
5.3	Chapter Abstract	41
5.4	Introduction	41
5.5	Methodology	43
5.5.1	Data	43

5.5.2	Geology and Soil Co-variates	43
5.5.3	Geo-FMs	44
5.5.4	Prediction Head	45
5.5.5	Experiment Protocols	45
5.6	Results and Discussion	47
5.6.1	Results	47
5.6.2	Discussion	49
5.7	Conclusion	50
6	Conclusion	53
6.1	Summary of the Results	53
6.2	Implications	54
6.3	Limitations	55
6.4	Future Directions	56

List of Figures

3.1	Curriculum learning diffusion framework: the despeckling difficulty is determined by the look value L , where single-look ($L = 1$) images represent the highest level of difficulty. . . .	11
3.2	Examples highlighting differences between (a) grayscale-converted RGB and (b) clean SAR images.	13
3.3	Image comparison showing our model (c), trained with curriculum learning and conditioned on a Lee filter. Unlike SAR-DDPM, our model effectively preserves the planes in the images.	15
3.4	Pixel distribution between the different datasets	18
4.1	The three stages of our workflow. The first, preprocessing , has a user prepare imagery and training data for ingestion by AI/ML models. Next, during the AI/ML stage, the user trains models to <i>detect</i> (find each target and report its location) and <i>segment</i> (draw the exact outline of each target to show its shape and area) imagery. The choice of model is up to the researcher. As part of the AI/ML stage, experts validate outputs of the trained model and produce refined uncertainties. A user may opt to retrain the model with these expert insights. The final stage, post-processing , involves merging output tiles and compiling metrics so as to produce a final product for downstream analysis.	23
4.2	Expert Verification GUIs. (a) False Positive (FP) Checker and (b) False Negative (FN) Checker. A progress bar shows how many tiles the expert has reviewed or skipped.	27
4.3	LFP detection applied to imagery of Kadwanchi village. (a) Raw 2019 satellite image and (b) predicted LFP masks on the village map (in white) from M_1 . Highlighted boxes on the map indicate the tiles shown in Figure 4.4a–c (cyan) and Figure 4.4d–e (orange).	32
4.4	Expert-validated model outputs on 2019 Kadwanchi imagery. (a) Original image tile. (b) Prediction from M_0 . (c) Prediction after expert verification and retraining (M_1). (d) Tile of the FN example, (e) Mask prediction of the FN example from (d). In this case, two closely spaced LFPs are recognized as a single entity, causing the unusually shaped LFP to be only partially detected by the model.	33
5.1	Training site sample image. Left: S2 true-color image. Right: PCA false color of per-pixel frozen OlmoEarth embeddings (PC1–PC3).	44

5.2	Temporal consistency of OlmoEarth Base embeddings via L_2 distance (left) and cosine similarity (right).	49
5.3	Effect of labeled training set size on test-set R^2 . Points show means across 30 runs; shaded bands are $\pm 1SD$	50
5.4	The geographic distribution of $k=15$ embedding clusters across 4,814 DS-801 sites. Each color corresponds to one cluster from k -means on L_2 -normalized OlmoEarth Base embeddings.	51
5.5	Effect of including the focal cluster training sites on nearby non-focal test sites (grouped into proximity zones). Shaded bands show ± 1 SD across seeds.	52
5.6	Absolute prediction error at SE coastal holdout sites for AI (lower is better). Maps compare the FM-only model (left) with one augmented with co-variates (right).	52

List of Tables

3.1	Despeckling on DSIFN Images (256×256, L=1 speckle)	16
3.2	Despeckling on SAR Images (256×256, L = 1 speckle)	16
3.3	Detection Performance Metrics	17
4.1	Detection performance and downstream task metrics on Kadwanchi images for two trained iterations.	34
S1	Glossary (methods-related technical terms)	38
S2	Specifications of satellite imagery datasets ^a	39
5.1	Soil element prediction performance. Values are the mean with standard deviation in parentheses. Bold values indicate the best result for each element.	48
5.2	CONUS gains from geology and SSURGO co-variates.	49
5.3	Held-out SE coastal performance with and without geology and soil co-variates.	50

1 Acknowledgments

I am grateful to my advisor, Akshay Mehra, for his guidance, patience, and support throughout my Ph.D. From the first day I walked into his office, his answer to my new projects has always been, “Let’s go for it,” encouraging me to think broadly and push beyond my limits. His feedback shaped both the direction of this dissertation and my development as a researcher.

I thank my committee members, Sameer Shah, Marine Denolle, David Butman, and Hayato Ushijima-Mwesigwa, for their time and advice on this work. I owe particular thanks to Hayato, who mentored me through my internships at Fujitsu and taught me how research gets done outside the university.

I thank David Catling, Jonathan Toner, Joshua Krissansen-Totton, and many others in the Astrobiology Program for their mentorship during earlier stages of my graduate training, which provided an important foundation for the work that followed.

I thank my friends and colleagues, Andrew Shumway, Lucas Fifer, Nicholas Wogan, Trent Thomas, Kimberly Sinclair, Shintaro Kadoya, Owen Lehmer, Sebastian Haas, and Sarah King, for many conversations, proofreading, and feedback throughout my time in the program. I also thank colleagues in the Department of Earth and Space Sciences and the Converging Technologies Lab at Fujitsu Research of America for comments on drafts, figures, presentations, and research ideas that contributed to this dissertation.

Finally, I thank my family for their unconditional love and support. My mom, Tzu-Ying, earned her own Ph.D. while raising two young kids, and thanked us in her dissertation’s acknowledgments; it is a joy, all these years later, to return the favor in mine. The nights she spent stargazing with us began the curiosity that led me to Earth and space science, and eventually to this degree. My dad, Wen-Bor, was always the first to step in when times were hard, carrying me through the challenges of graduate school without ever being asked. And my brother, Ping-Min, taught me many things and graciously gave me a place to stay so I could survive the Bay Area during my internships.

This work was supported by the Department of Earth and Space Sciences and the School of

Environmental and Forest Sciences at the University of Washington, including research assistantship support through the UW Global Innovation Fund. Chapter 1 grew out of a research internship at Fujitsu Research of America.

2 Introduction

2.1 Background

Landscapes change constantly through floods, wildfires, and urban development, and tracking these changes matters for science, risk management, and policy. Satellite imagery has supported landscape observation for decades: for example, synthetic aperture radar (SAR) imagery provides all-weather, day-and-night imaging and is especially useful when optical imagery is limited by cloud cover. Likewise, high-resolution commercial optical satellites provide sub-meter imagery that can reveal highly detailed landscape features. But the volume, variety, and detail of satellite imagery have grown rapidly in recent years. Critically, the growing data archives quickly become too large to analyze manually, making artificial intelligence (AI) and machine learning (ML), hereafter AI/ML, central to extracting information from these observations.

Much of this AI/ML capability comes from computer vision, the field of algorithms that extracts information from images. Deep learning models now achieve high benchmark performance on image-analysis tasks such as object detection and segmentation (Minaee et al., 2021; Zou et al., 2023). Many of these models are released as pretrained weights or software packages, so researchers can adapt an existing high-performing model to satellite imagery rather than building one from scratch for each task. This availability has changed the role of AI/ML in remote sensing research, where computer vision applications have become reusable workflows rather than one-off, task-specific models.

Together, the increasing influx of data and advances in modeling have changed how Earth and environmental researchers analyze information for science and decision-making (Huntingford et al., 2019; Rolnick et al., 2022). AI/ML model outputs now inform analyses ranging from landscape changes to environmental impacts on marginal communities, as practitioners and policymakers use these applications to allocate resources and assess risks to infrastructure (Bostrom et al., 2024). As AI/ML model outputs move closer to public-facing decisions, their value depends on the user's

ability to judge an output’s meaning.

2.2 The Gap in Geospatial AI Modeling

Scientific and policy decisions based on geospatial data rarely depends on a single predicted value or one model score. Judging reliability requires more specific information, which includes the kinds of errors a model makes, the conditions under which those errors occur, and the effect of the resulting uncertainty on the downstream analysis. In standard AI/ML model development, however, this information is often reported in technical terms that are difficult for users who do not necessarily have experience in AI/ML (Rolf, 2023; Wagstaff, 2012).

Within an AI/ML model development workflow, technical metrics remain useful for tracking progress and comparing methods. In remote sensing, for example, image-quality and detection metrics can rank model outputs, but a higher score does not show whether an output preserves the information needed for the downstream task or introduces bias into the final estimate. Before these results can inform scientific or decision-making tasks, they must be translated into terms that users can connect to the target question (Devillers et al., 2010).

I refer to the missing or incomplete connection between model-centered development and task-specific use as a “reliability gap” in geospatial AI. This gap appears when technical metrics, model outputs, or pretrained tools are treated as sufficient evidence for developers that a model is reliable for the intended scientific task, before that evidence has been translated into the task-specific information for users. When this translation is missing, validation of the AI/ML workflow outputs is incomplete. A model may score well on its benchmarks on a carefully curated training set, but users have little justification for treating the output as reliable evidence, which limits how effectively AI/ML workflows can support scientific analysis and decision-making.

2.3 Designing Reliability into Geospatial AI Workflows

Reliability is a core requirement for trustworthy AI systems (National Institute of Standards and Technology, 2023). The NIST Artificial Intelligence Risk Management Framework (AI RMF 1.0)

lists validity and reliability first among the characteristics of trustworthy AI and treats them as the base on which the other characteristics, including safety, transparency, and fairness, are built. The framework is written to apply across multiple sectors in general, so it defines what a reliable system should achieve but leaves how to achieve the goals to each application domain. In the geospatial domain, one aspect of reliability, the replicability of findings across space and time, is already known to be limited. For example, Goodchild and Li (2021) argue that findings in geographic research are hard to replicate. The primary reason is spatial heterogeneity, meaning the physical or statistical properties differ from place to place and over time, so a relationship estimated in one setting may not hold in another. Replicability is further limited by choices that tie a result to its original assumptions, such as how variables are defined or how scales are represented. Reliability in geospatial AI, therefore, has to be established within the domain, either through design choices that account for the behavior of geospatial data and processes, or through documentation transparent enough that users can see how a result depends on those choices.

The question this dissertation aims to answer is how reliability can be built into geospatial AI workflows so that model outputs remain aligned with the scientific tasks they are meant to support. Simply put, reliability means that a workflow produces outputs that fit the intended scientific task, can be interpreted by users, and includes enough validation or uncertainty information to support scientific judgment. This framing treats interpretability and uncertainty as design requirements for the model developer, not only as post hoc explanations after training. Meeting these requirements is a property of the whole workflow rather than of any single stage (National Institute of Standards and Technology, 2023).

Ultimately, when a geospatial AI output is used in scientific analysis, its reliability depends on more than the computer vision model architecture alone. The reliability of an AI/ML application also depends on choices made before and after training, from input preparation and training strategies to output validation. Any of these steps can change the meaning of the final result for a user. For example, a preprocessing step may remove task-relevant information, or a validation procedure may leave users without enough context to judge the output. Evaluating a geospatial AI

application, and its true “performance,” therefore, requires examining the full process that turns remote sensing data into scientific information.

2.4 Expert Guidance in the Geospatial AI Workflow

Having expert guidance, hereafter referred to as “guidance”, is the key to designing reliability into geospatial AI workflows. Guidance refers to domain knowledge that shapes the training, checking, and interpretation of an AI/ML system so that it aligns with the downstream tasks (Von Rueden et al., 2021). Guidance under this definition is active rather than reactive. Conventional evaluation acts as a guardrail, judging results only after the model output is produced, whereas guidance embeds domain knowledge into the workflow stages themselves (Karpatne et al., 2017; Willard et al., 2022). In a typical AI/ML development cycle, evaluation comes after the model is trained and deployed, using metrics that do not necessarily align with the user’s perspective. If the model shows bias, users would want to know which part of the workflow contributes to the bias. If the data are biased before training, the evaluation setup in earlier phases could correct the data or clarify the effects of that bias before it enters the model weights.

In some cases, guidance on the data preprocessing stage is beneficial by structuring the inputs or the learning objective around the information that must be preserved in the data. In other cases, guidance enters through validation, where frameworks such as human-in-the-loop with a standardized protocol allow users to identify and document errors that benchmark metrics may miss, and also help define the metrics that are useful for the task (Gil et al., 2019; Mosqueira-Rey et al., 2023). Guidance can also come from the environmental context when the model is applied to a prediction problem where imagery alone does not contain all the information needed for the scientific question. Although different types of data and objectives lead to different ways to incorporate guidance, the purpose of these methods is to align model behavior with the downstream task before the output is used as evidence for analysis or decisions. This gives users a clear basis for confidence because reliability is evaluated in terms that match the application.

In this dissertation, I apply “guidance” at multiple stages of the workflow, with multiple types

of geospatial data and models, to demonstrate how guidance aligns model outputs with the tasks they support in downstream analysis. Each chapter incorporates guidance into a different stage of model development or evaluation and measures the effect of that guidance on the output for the scientific task. Chapter 1 focuses on the image-enhancement stage, where guidance is built into the training strategy so that SAR despeckling preserves the information needed for downstream object detection. Chapter 2 moves to the validation stage, providing an iterative and transparent framework in which expert review translates object-detection outputs into task-relevant metrics and uncertainty. Chapter 3 extends the idea of reliability to geospatial foundation models, testing whether their soil-chemistry predictions remain consistent across space and time and whether prior knowledge, such as geology, can guide those predictions under sampling constraints. These three chapters show that model outputs become scientifically useful when workflows are guided by downstream task performance, expert review, and environmental context, and are evaluated in terms of the downstream task objectives. Designing reliability into the workflow from the start turns model outputs into evidence that users can interpret and act on.

3 Chapter 1: Expert Guided Diffusion for Synthetic Aperture Radar Despeckling

3.1 Relevance

Chapter 1 addresses the image-enhancement stage of the geospatial AI workflow, where a preprocessing step can remove task-relevant information before analysis begins. We incorporate guidance into the training strategy through a curriculum-guided diffusion framework that conditions the model on numerical speckle filters and organizes training by noise intensity. This design mitigates speckle noise without erasing features needed for downstream object detection, showing how guidance built into the despeckling model itself keeps this preprocessing step aligned with the scientific task.

3.2 Status

This chapter is adapted from work published as Lin, P.-C., Ushijima-Mwesigwa, H., & Au, W. Y. (2025). Expert-guided diffusion for synthetic aperture radar despeckling. *Proceedings of the 33rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. The chapter extends the published version with an expanded literature review; the methods and results are unchanged.

3.3 Chapter Abstract

Synthetic Aperture Radar (SAR) images are critical for remote sensing but are degraded by speckle noise, which hampers visual quality and downstream tasks. Traditional despeckling methods often over-smooth or hallucinate features, especially on out-of-distribution data. We propose a Curriculum-Guided Diffusion Framework that combines statistical priors with a data-driven model, using curriculum learning across speckle levels and conditioning on numerical filters. A cascaded

diffusion strategy further boosts performance on real SAR images, even when trained on synthetic data. Experiments show improved PSNR, SSIM, and object detection, making our approach robust and deployable for real-world SAR applications.

3.4 Introduction

Synthetic Aperture Radar (SAR) data are produced through an active sensing process that captures radar signals and reconstructs them into high-resolution images. Unlike optical imagery, which offers rich spectral information for tasks like land cover classification, SAR excels in capturing structural and geometric details of the Earth’s surface. SAR operates independently of daylight and weather, providing consistent, all-weather imaging capabilities, making it especially valuable for time-sensitive geospatial applications in challenging environmental conditions.

One of the main artifacts impacting SAR image quality is *speckle*, a granular noise pattern inherent to the imaging process. As described in Goodman (1976), speckle arises from the coherent interference of radar signals reflected by multiple scatterers within a single resolution cell. This phenomenon reduces image contrast and obscures fine spatial details, making interpretation and analysis more difficult (Dalsasso et al., 2022). Consequently, effective speckle reduction is essential to enhance SAR imagery and ensure the accuracy and reliability of downstream geospatial applications, including surveillance, disaster response, infrastructure assessment, and surface change detection (Mondini et al., 2021; Tay et al., 2020; T. Zhang et al., 2021).

Spatial numerical filters have a long history in SAR despeckling. Traditional methods like the Lee and Kuan filters reduce speckle noise by averaging pixel values within a local window, guided by local mean and variance estimates (Kuan et al., 1985; Lee, 1981; Paul & Savakis, 2025). In addition, non-deep learning methods such as non-local means (NLM) and BM3D remain widely used due to their generalizability across datasets and their lack of dependency on training (Parrilli et al., 2011). Despite these advantages, these approaches often oversmooth the data, leading to the loss of fine image details and structural information.

Deep learning methods like SAR-CNN, ID-GAN, and diffusion models (DDPMs) leverage

spatial context for despeckling (Chierchia et al., 2017; Perera et al., 2023). While effective, these methods may hallucinate or erase features, risking errors in tasks like object detection and segmentation (Ko & Lee, 2021; Paul & Savakis, 2025). To address these challenges, we propose a noise-level-guided framework designed to reduce hallucination artifacts and improve detection accuracy in SAR despeckling tasks. Our approach adopts a cascaded diffusion strategy inspired by successful applications in image super-resolution, where multiple stages progressively enhance the image quality (Ho et al., 2022). Additionally, we incorporate curriculum learning (Bengio et al., 2009), a training technique that organizes data by increasing difficulty. This human-inspired strategy accelerates model convergence and enhances generalization, making it easier to train deep networks on complex SAR datasets.

The main contributions of this chapter are:

1. We introduce a Curriculum-Guided Diffusion Framework for SAR despeckling that balances visual quality and structure preservation.
2. We incorporate statistical priors from traditional numerical filters into the diffusion process to enhance generalization, particularly on out-of-distribution data.
3. We introduce a cascaded diffusion architecture that achieves strong despeckling performance on real SAR images, even when trained only on synthetically speckled grayscale RGB data.
4. We conduct comprehensive experiments on both synthetic and real-world SAR data to validate our approach.

3.4.1 Speckle Noise Model

A speckled SAR image Y is modeled as the product of the underlying clean reflectivity image X and a noise term N :

$$Y = XN \quad (3.1)$$

The speckle component N can be described by Gamma distribution with unit mean and a

variance inversely proportional to the number of looks L when acquiring the image. We often use this equation to describe the effect:

$$p(N) = \frac{1}{\Gamma(L)} L^L N^{L-1} e^{-LN} \quad (3.2)$$

This multiplicative and signal-dependent nature of speckle makes it different from Gaussian noise for typical images, hence the bias correction is potentially different from typical image denoising algorithms.

3.4.2 Diffusion Models

Diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015) have emerged as a powerful class of generative models, achieving state-of-the-art performance in image synthesis and often surpassing GANs in visual fidelity. These models have been successfully applied in both unconditional and conditional generation tasks. Beyond traditional image synthesis, their applications now span across audio and video generation, segmentation, language modeling, and more recently, SAR image despeckling. In this work, our model is based on the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020). For a detailed overview of DDPM, we refer the reader to the original work.

A Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) defines two Markov processes: a forward diffusion process that gradually corrupts input data by adding Gaussian noise, and a reverse generative process that learns to denoise and recover samples from pure noise. Given input data $x_0 \sim q(x_0)$, the forward process generates a sequence $\{x_t\}_{t=1}^T$ via:

$$q(x_1, \dots, x_T | x_0) = \prod_{t=1}^T q(x_t | x_{t-1}), \quad (3.3)$$

with Gaussian transitions defined as:

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t \mathbf{I}), \quad (3.4)$$

where $\beta_t \in (0, 1)$ controls the noise schedule.

To sample new data, the reverse process starts from a Gaussian prior $x_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoises using a learned distribution:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad (3.5)$$

where μ_θ and Σ_θ are predicted by a neural network trained to approximate the reverse transitions. This iterative denoising process allows diffusion models to generate high-quality samples that closely match the original data distribution.

3.4.2.1 Cascaded Diffusion Models Cascaded diffusion models have emerged as a powerful strategy in image generation tasks, particularly in scenarios requiring high-resolution outputs. Unlike single-stage diffusion models that perform denoising over a fixed noise schedule, cascaded approaches decompose the generation process into multiple stages, where each stage progressively refines the output at increasing resolutions or varying noise levels (Ho et al., 2022). This design improves sample quality and stability, allowing the model to focus on coarse-to-fine details in a structured manner. Cascaded diffusion frameworks have demonstrated strong performance in tasks such as super-resolution, inpainting, and high-resolution synthesis, and are especially effective when combined with conditioning mechanisms or guidance strategies. We adopt a similar cascaded diffusion framework for SAR despeckling.

3.4.2.2 Curriculum Learning Curriculum learning is a training strategy that presents samples in order of increasing difficulty, accelerating convergence and improving generalization (Bengio et al., 2009). This training strategy imitates the learning process of humans, by classifying data into different categories that resemble difficulty. By starting to train from the easiest difficulty, the model tends to converge to the optimum faster, which makes training deep learning models on complex datasets easier.

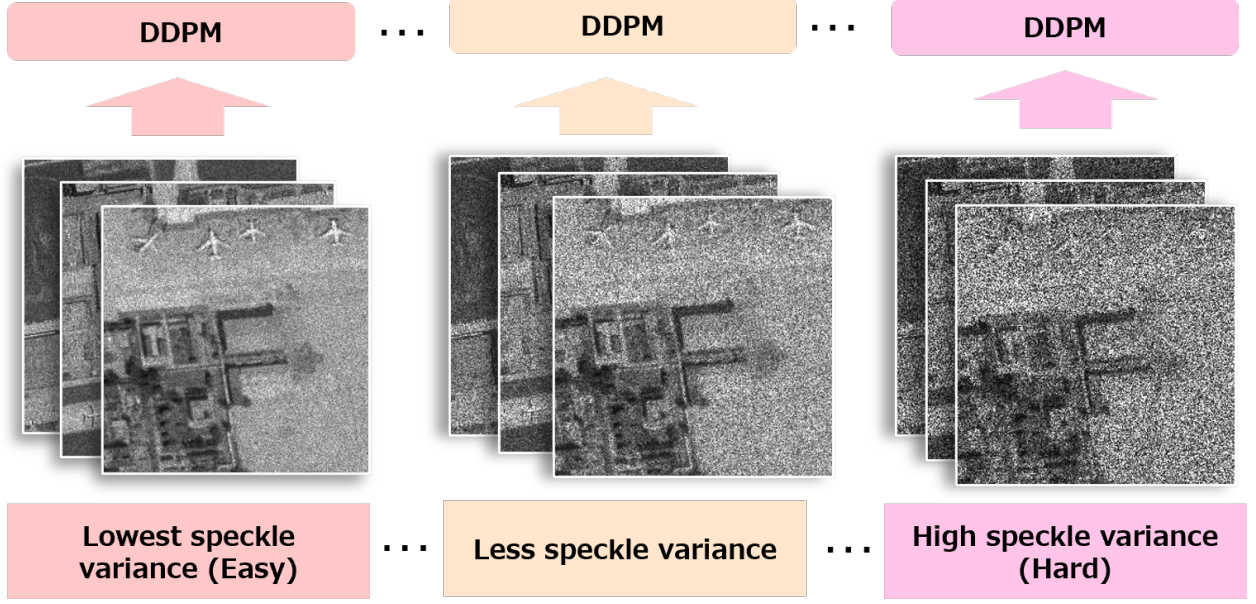


Figure 3.1: Curriculum learning diffusion framework: the despeckling difficulty is determined by the look value L , where single-look ($L = 1$) images represent the highest level of difficulty.

3.5 Methodology

3.5.1 SAR-DDPM and the Loss Equation

We perform SAR despeckling by extending conditional SAR-DDPM (Perera et al., 2023) with a U-Net noise predictor conditioned on speckled inputs and self-attention in residual blocks. Following (Nichol & Dhariwal, 2021; Paul & Savakis, 2025), we train using the improved DDPM objective, whose per-step loss is

$$Loss_{\text{total}} = Loss_{\text{simple}} + Loss_{\text{vb}}. \quad (3.6)$$

where the mean prediction loss, $Loss_{\text{simple}}$, measures the squared error between the true noise ϵ and the noise predicted by the network $\epsilon_{\theta}(\mathbf{x}_t, t)$:

$$Loss_{\text{simple}} = E_{t, \mathbf{x}_0, \epsilon} [\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, t)\|^2] \quad (3.7)$$

where $\mathbf{x}_t$ is the noisy image at timestep t , derived from the clean image $\mathbf{x}_0$ and the sampled Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The variance learning loss, $Loss_{vb}$, trains the model to predict the variance $\Sigma_\theta(\mathbf{x}_t, t)$ of the reverse diffusion step. It is formulated as the KL divergence between the true reverse posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and the model’s predicted reverse distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$:

$$Loss_{vb} = E_{t, \mathbf{x}_0, \epsilon} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))] \quad (3.8)$$

For $t = 0$, $Loss_{vb}$ is replaced by the negative log-likelihood of the data term.

3.5.2 Training Framework

Our speckle-level guided framework (Section 3.4.2.2) ranks difficulty by L in Eq. (3.2): higher L (multi-look) images have less speckle and retain more structural detail, making them easier (e.g., $L = 10$), while $L = 1$ (single-look) images are maximally speckled and most challenging. We generate training data across $L = 10 \rightarrow 1$ and evaluate on the hardest level ($L = 1$). In curriculum learning with $n = 10$ levels, we introduce level $L = n - 1$ every i iterations until reaching $L = 1$ (Figure 3.1).

Another strategy is a two-stage cascaded DDPM (Ho et al., 2022), which splits despeckling at an intermediate speckle level $L = n$. Stage 1 is trained to map images from $L = 1$ (single-look) to $L = n$ (multi-look), and Stage 2 from $L = n$ to the clean (ground truth) image. At inference, an $L = 1$ input is first passed through Stage 1 to produce an $L = n$ output, then through Stage 2 for full despeckling (e.g. for $n = 4$: Stage 1: $1 \rightarrow 4$, Stage 2: $4 \rightarrow \text{clean}$).

Besides speckle conditioned inputs, we optionally concatenate externally generated Lee-filtered images into the DDPM. Users can select the filter window size and speckle level for desired smoothness. Under a mixed- L curriculum, each speckle level $L = n$ is paired with its corresponding $L = n$ filtered image during both training and inference.

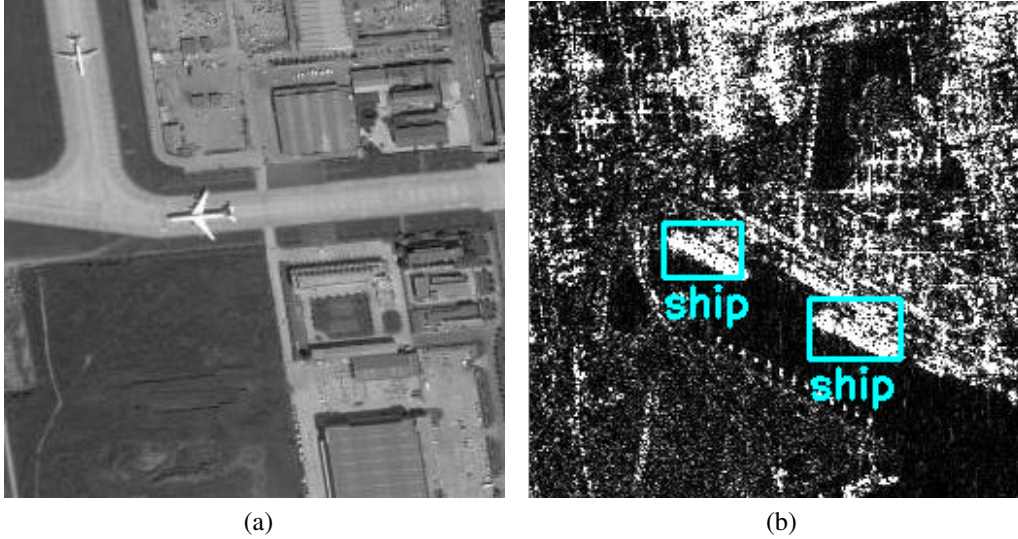


Figure 3.2: Examples highlighting differences between (a) grayscale-converted RGB and (b) clean SAR images.

3.6 Experiments

3.6.1 Training Data

Many despeckling models use clean grayscale images derived from RGB data as ground truth and synthetically add speckle noise to simulate SAR images. These synthetic images are often called "pseudo-ground truth" to emphasize that they approximate the underlying clean SAR signal, either through simulation or by partial despeckling. In this study, we experiment with both synthetic and real SAR data for training and inference. We use the DSIFN dataset, an RGB satellite dataset widely adopted for synthetic SAR generation in despeckling studies (Perera et al., 2023; C. Zhang et al., 2020)

To evaluate despeckling and detection performance on real SAR images, we use the SARDet100k dataset for inference. SARDet100k consists of $\sim 120,000$ SAR images from multiple platforms, including Sentinel-1, TerraSAR-X, Gaofen-3, and airborne SAR, covering C-band and X-band observations. The dataset includes COCO-style bounding box labels for six object classes: ships, aircraft, vehicles, bridges, harbors, and tanks (Li et al., 2024).

3.6.2 Experiments

We conducted two experiments: one benchmarks image quality using peak signal-to-noise ratio (PSNR) and structured similarity index (SSIM) on despeckled images defined in literature (Paul & Savakis, 2025), and the other evaluates detection performance using a YOLOv8 model trained on SARDet-100k to assess any changes after despeckling.

In the first experiment, we train models using several strategies from Section 4.5.2: curriculum learning ($L = 10 \rightarrow L = 1$), two-stage cascaded DDPM ($L = 1 \rightarrow L = 4$; $L = 4 \rightarrow \text{Clean}$), and filter-conditioned DDPM. We convert 6,305 DSIFN RGB images to grayscale following Perera et al. (2023), and apply speckle noise using Eq. (3.1) and (3.2), generating speckle levels from $L = 10$ to $L = 1$ to simulate varying speckle in labeled SAR datasets. Besides generating speckles for the grayscaled-converted RGB images, we also generate filtered images for conditioning using a Lee filter with a 7 pixel window size, to guide DDPM training.

The training curriculum begins from the least speckled level ($L = 10$), adding L_{n-1} every 375 iterations. Each strategy trains a DDPM model for 8,000 iterations on a single NVIDIA H100 GPU with a learning rate of 0.0001 and batch size 16. We also conduct ablation studies on the curriculum learning model, including: baseline SAR-DDPM with only $L = 1$ speckle (Perera et al., 2023), random speckle levels between $L = 1$ and $L = 10$, and inverse difficulty ($L = 1 \rightarrow L = 10$, with new levels added every 375 iterations).

After training, we perform inference on a validation set of 100 random DSIFN images. The same model weights are then applied to SARDet100k images. Prior to inference, we select a subset containing ship labels and less than 50% zero-valued pixels, as low-value regions mostly correspond to background like the ocean. We add $L = 1$ speckle to 100 random SARDet100k images and apply the trained weights to directly despeckle them.

The second experiment evaluates detection performance on real SAR images from different learning strategies. To simplify the task, we focus on ship detection, as shown in Figure 3.2b. The detection model was trained on 46,000 images of size 256×256 for 100 epochs.

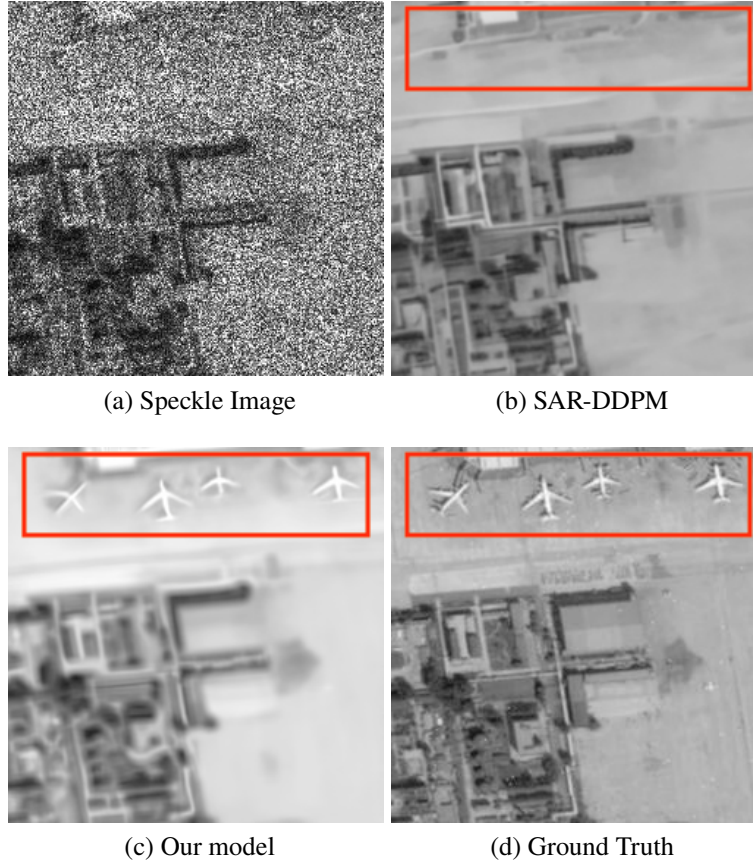


Figure 3.3: Image comparison showing our model (c), trained with curriculum learning and conditioned on a Lee filter. Unlike SAR-DDPM, our model effectively preserves the planes in the images.

3.7 Results and Discussion

3.7.1 Results

Table 3.1 presents the despeckling results on 100 DSIFN images. The model trained with a random curriculum using $L = 1$, $L = 10$ speckle achieves the highest PSNR, while the curriculum-trained DDPM conditioned with a Lee filter yields the highest SSIM.

Figure 3.3 compares a speckled image, an image despeckled by SAR-DDPM (baseline), and one despeckled by our DDPM with curriculum learning and Lee filter conditioning. Our model preserves critical details (e.g., aircraft) necessary for detection, whereas the SAR-DDPM baseline erases the aircraft in its output.

Table 3.2 shows the despeckling results on 100 SAR images from SARDet100k. The model trained with curriculum achieved the highest PSNR, while the cascaded model yielded the highest SSIM.

Table 3.1: Despeckling on DSIFN Images (256×256, L=1 speckle)

Method	PSNR (dB)	SSIM
SAR-DDPM (Baseline)	23.730	0.603
DDPM with filter	19.006	0.583
Cascaded	16.732	0.301
Random Curriculum	23.772	0.611
Reverse Curriculum	23.329	0.588
Curriculum	23.397	0.592
Curriculum & filter	18.882	0.657

Table 3.2: Despeckling on SAR Images (256×256, L = 1 speckle)

Method	PSNR (dB)	SSIM
SAR-DDPM (Baseline)	25.852	0.650
DDPM with filter	15.471	0.219
Cascaded	26.755	0.817
Random Curriculum	26.138	0.646
Reverse Curriculum	24.811	0.515
Curriculum	26.821	0.741
Curriculum & filter	15.816	0.356

Table 3.3 presents detection results on labeled SAR images using a YOLOv8 model trained for ship detection on SARDet100k. On clean images, the model achieves an mAP_{50-95} of 0.6181, mAP_{50} of 0.9387, and F1 score of 0.8934. After adding $L = 1$ speckle, performance drops to $\text{mAP}_{50-95} = 0.4802$, $\text{mAP}_{50} = 0.7887$, and F1 score = 0.7219.

3.7.2 Discussion

Table 3.2 and Table 3.3 show that training with mixed noise levels yields the best overall performance. These results suggest that exposure to lower noise levels during training enhances the model’s ability to preserve fine details and smooth out-of-distribution data. For real SAR

Table 3.3: Detection Performance Metrics

Image	mAP ₅₀₋₉₅	mAP ₅₀	F1 Score
SAR-DDPM (Baseline)	0.5390	0.8678	0.8361
Cascaded	0.5687	0.8786	0.8415
DDPM with filter	0.2695	0.5590	0.5462
Random Curriculum	0.5579	0.8834	0.8400
Reverse Curriculum	0.4735	0.8353	0.8079
Curriculum	0.5882	0.8943	0.8603
Curriculum & filter	0.3606	0.7716	0.7459

despeckling and detection, curriculum learning outperforms other strategies, demonstrating that starting with easier targets leads to better results.

Table 3.1 and Table 3.2 show that models excelling on DSIFN do not necessarily perform best on real SAR. The curriculum-learning model, although not attaining the highest PSNR or SSIM, outperforms SAR-DDPM on real SAR (Table 3.2 and Table 3.3). Domain shift between synthetic SAR (from RGB) and real SAR likely explains these discrepancies. Figure 3.4 shows that real SAR pixel values in SARDet100k are on average lower than synthetic SAR from DSIFN, reflecting the darker oceanic backgrounds typical in ship-detection scenarios versus grayscale images derived from RGB datasets.

Figure 3.3 shows the output of the DDPM conditioned with a Lee filter (window size 7). This qualitative comparison demonstrates that the conditioning preserves important details (e.g., aircraft) that are lost by a standard SAR-DDPM.

From our experiments, conditioning with numerical filters retains fine structural detail in DSIFN despeckled images and preserves object shapes under heavy speckle, but introduces brightness inconsistencies that reduce image consistency, smoothness, PSNR, and SSIM on real SAR data. Thus, applications must balance image fidelity against detection performance, and future work could explore adaptive conditioning strategies to optimize this trade-off.

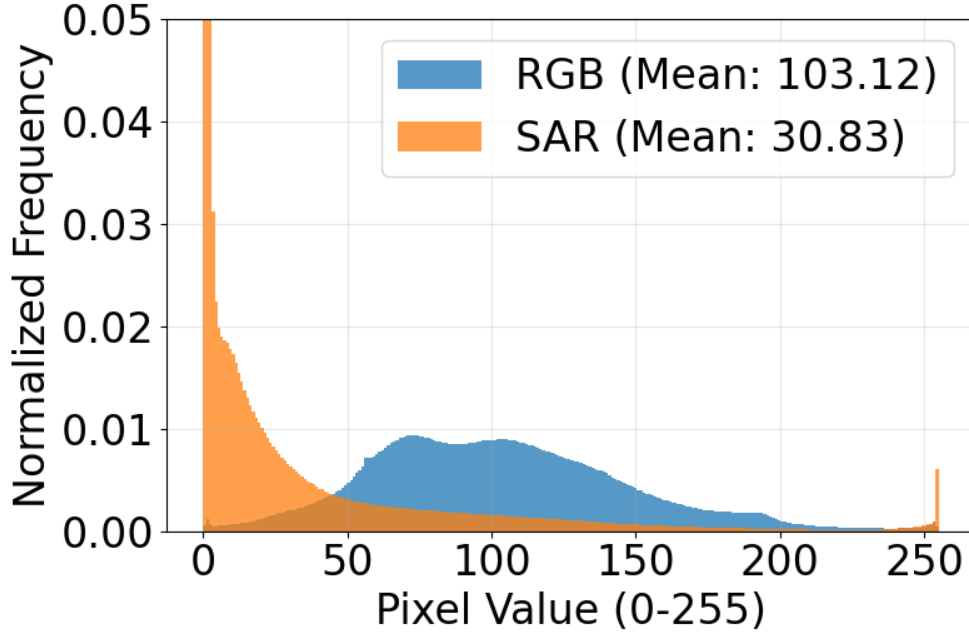


Figure 3.4: Pixel distribution between the different datasets

3.8 Conclusion

This study presents noise-guided training strategies to improve SAR-DDPM despeckling. By combining mixed-noise level training with a curriculum-learning schedule, we achieve notable gains in image quality and object detection, particularly on out-of-distribution data. Ablation studies reveal curriculum learning outperforms random or reverse orderings, and incorporating numerical-filter conditioning helps retain structural details while balancing PSNR, SSIM, and detection accuracy. Altogether, the integration of diffusion-based denoising, mixed-look training, and filter-based guidance enables robust despeckling across both synthetic and real SAR datasets.

4 Chapter 2: Incorporating expert inputs to produce calibrated outputs: Building confidence in applications of machine learning to remote sensing

4.1 Relevance

Chapter 2 addresses the validation stage of the geospatial AI workflow, where a validation procedure can leave users without enough context to judge model outputs. I build guidance into this stage by establishing an iterative and transparent protocol for expert validation, in which domain experts review object-detection outputs over repeated rounds, and their feedback converts model predictions into task-relevant metrics and uncertainty. The protocol gives expert review a repeatable structure within the AI/ML pipeline, so the validation process itself can be documented, inspected, and reused. Applied to lined farm pond detection in Kadwanchi, India, this workflow translates benchmark-level results into the task-specific information users need before treating the outputs as evidence for downstream analysis.

4.2 Status

This work has been submitted to *AGU Earth and Space Science* and is currently under review. Lin, P.-C., Mehra, A., & Shah, S. H. (under review). Incorporating expert inputs to produce calibrated outputs: Building confidence in applications of machine learning to remote sensing. *Earth and Space Science*

4.3 Chapter Abstract

Artificial intelligence and machine learning (AI/ML) applications that analyze features (e.g., crops or infrastructure) in remotely sensed data often output technical evaluation metrics that do not directly translate into real-world uncertainties. Because policy- and decision-makers increasingly

rely on AI/ML outputs to guide actions, such as allocating resources or assessing environmental change, this misalignment can reduce model effectiveness and lead to poorly informed actions. To enhance confidence in the use of AI/ML for geospatial research, we propose a framework that: i) formalizes how subject-matter experts verify and refine models using an interactive interface and well-documented review procedure, and ii) outputs decision-ready metrics for use by end-users, including policy- and decision-makers. We demonstrate the effectiveness of our approach by applying it to the identification and quantification of lined farm ponds (LFPs) in Maharashtra, India. On data that were not seen during training, expert verification increased the fraction of LFPs that the model correctly identified from 42% to 88% out of all LFPs present in the imagery and reduced counting and area estimate errors (from -58% to -12% and -52% to -7% , respectively). Overall, our work serves to highlight the power of expert validation and uncertainty calibration in producing AI/ML applications that end-users can use with greater confidence.

4.4 Introduction

Artificial intelligence and machine learning (AI/ML) have the potential to transform the analysis of Earth observation data across a wide range of domains, including agricultural sciences, urban planning, climate adaptation, and disaster response (e.g., Huang et al., 2018; Huntingford et al., 2019; Kamilaris & Prenafeta-Boldú, 2018; Rolnick et al., 2022). Such uses of AI/ML require care, both in the development and deployment of applications, and especially in cases where outputs might influence policy decisions that affect individuals' everyday lives. Recognizing this fact, AI/ML experts have begun to disseminate scholarship on what “trustworthy” and “confident” AI might mean (e.g., Bostrom et al., 2024; Herrera-Poyatos et al., 2025; Kaur et al., 2022; McGovern, Bostrom, et al., 2022; McGovern, Ebert-Uphoff, et al., 2022).

Fundamentally, confidence in AI/ML applications derives from a user's understanding of the limitations and biases of utilized techniques (Bostrom et al., 2024). Often, AI/ML models output technical metrics—such as mean average precision (mAP), intersection-over-union (IoU), precision, and recall in the case of object-detection tasks (Ghorbanzadeh et al., 2022; Tundia et al., 2023; Xia

et al., 2023, see Table S1 for definitions). These metrics may have limited relevance or meaning for practitioner audiences. Additionally, such metrics are not comprehensive descriptors of uncertainty and reliability (Bolya et al., 2020). For instance, while a high IoU score indicates a significant overlap between a prediction and a ground truth, by itself, it does not tell the user whether a model *systematically* under- or over-estimates the size of features (Y. Zhang et al., 2021). This limitation can give users false confidence in the model’s estimates and lead to biased decisions. In contrast, confidence intervals on estimates of area can provide users with a more intuitive—and, therefore, actionable—description of reliability. Providing such metrics is a form of “calibration”, which we define as the translation of internal measures into task-specific metrics that enable users to evaluate model performance for downstream tasks (Devillers et al., 2010). Here, downstream tasks refer to the practical analyses or decisions users want to make with the model outputs, such as assessing surface change over time. Such uncertainties make an application’s outputs explicit, actionable and aligned with downstream tasks for interdisciplinary research. Note that the act of “calibrating” uncertainty by producing task-specific metrics is different from a statistical calibration of uncertainty, which requires its own approach (e.g., Section 4.5.4.1).

The integration of domain experts can increase user confidence in an AI/ML application. Such *human-guided* approaches enable subject-matter-experts to participate in data preparation, model selection, and iterative refinement of predictions, thereby providing instant feedback that can improve performance (see also Gil et al., 2019; Mosqueira-Rey et al., 2023, for a review on recent human-guided methods to improve model predictions). For example, an expert might identify errors in a trained model’s predictions, which can be used to retrain the model and ensure that it does not make the same mistakes in the future.

The use of expert input in remote sensing is not novel. Indeed, many existing studies incorporate reviews via visual inspection or manual correction of model outputs. However, descriptions of these approaches are often ad-hoc or written without clear review guidelines and enough documentation for others to reproduce the process (Stehman & Foody, 2019; Tyukavina et al., 2025). These properties are counter to existing remote sensing validation guidance (e.g., Committee on

Earth Observation Satellites, see Tyukavina et al., 2025), which emphasizes that practitioners plan rigorous accuracy assessment in advance, use an explicit review design, and provide transparent documentation for independent verification (Olofsson et al., 2014; Stehman & Foody, 2019; Tyukavina et al., 2025).

Here, we introduce a framework that formalizes expert review, iterative model refinement, and task-aligned uncertainty reporting via a transparent validation workflow for geospatial AI/ML. Rather than reporting model confidence alone, our framework expresses uncertainty in terms that are directly usable for downstream analysis. We demonstrate our approach by applying it to the detection of agricultural features in Maharashtra, India, illustrating both improved model performance and the generation of actionable metrics for downstream tasks.

4.5 Methodology

4.5.1 Overall Framework

Our workflow follows a three-part structure: (i) preprocessing, (ii) AI/ML training, and (iii) post-processing (Figure 4.1). We provide a “plug-and-play” Python package of this workflow (see Section 4.9), allowing others to adapt it for their own purposes. Below, we provide details on each stage of the framework.

4.5.2 Preprocessing

The first step of developing any AI/ML application is to generate training data. In the case of imagery (e.g., satellite-derived), such data are masks: binary images where ones correspond to pixels that are part of an object and zeros to everything else. Masks represent the idealized target output of a model. Our workflow accepts masks produced by any tracing software (e.g., Roboflow, LabelMe, or Adobe Photoshop) that outputs a binary raster in a standard file format (e.g., Portable Network Graphics or Joint Photographic Experts Group [.PNG or .JPEG]). Our workflow requires both the mask and the original imagery data from which it was derived. Each “image-mask pair”

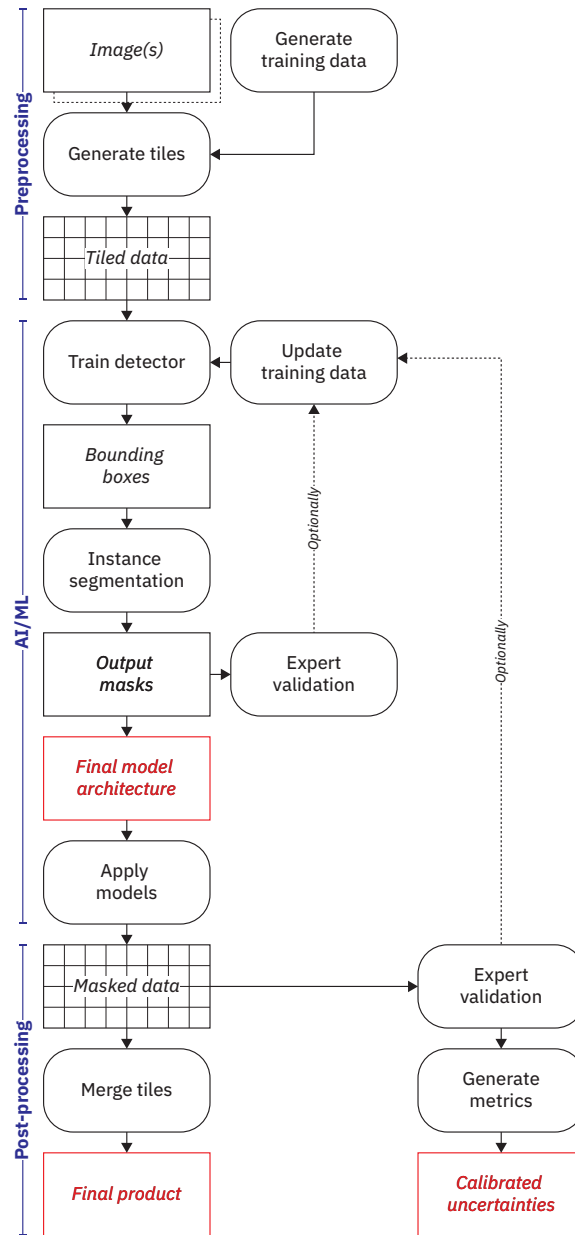


Figure 4.1: **The three stages of our workflow.** The first, **preprocessing**, has a user prepare imagery and training data for ingestion by AI/ML models. Next, during the **AI/ML** stage, the user trains models to *detect* (find each target and report its location) and *segment* (draw the exact outline of each target to show its shape and area) imagery. The choice of model is up to the researcher. As part of the AI/ML stage, experts validate outputs of the trained model and produce refined uncertainties. A user may opt to retrain the model with these expert insights. The final stage, **post-processing**, involves merging output tiles and compiling metrics so as to produce a final product for downstream analysis.

must share the same dimensions and alignment, which is easiest to achieve by first cropping imagery around a feature of interest, saving that cropped file, and then using it to create the corresponding mask. Our workflow includes scripts to align, crop, and organize image-mask pairs.

All image-mask pairs must have the same shape for our pipeline to process them consistently. The workflow enforces this constraint by adding background pixels around image/mask borders (i.e., padding). The workflow then divides padded image-mask pairs into tiles (i.e., tiling) using a window of a given dimension (i.e., a kernel) and a step size of how many pixels to move over between tiles (i.e., stride). Our naming convention for tiles retains the coordinates of the top left pixel of the data, enabling users to easily re-merge the tiles or find a tile from a set of coordinates. Next, the workflow excludes tiles where the images comprise primarily background (e.g., $> 50\%$ pixels set to 0; this threshold is user-adjustable), and splits the tiles into training and validation subsets via random selection. During training, the validation subset—which is not used for fitting model parameters—provides an estimate of how the model performs when applied to previously unseen data.

The final step before training a model involves creating or assigning labels for each tile. Most existing object detection or segmentation algorithms do not directly ingest the pixels of mask tiles, but, rather, require text files called “labels” that describe the location and/or shapes of the feature of interest in each tile. For detection tasks (see Section 4.5.3.1.1), labels typically are the four corners, or the dimensions, of a mask’s bounding box (bbox). For segmentation (see Section 4.5.3.1.2), labels are either text that records the coordinates of an outline around each object or pixel-by-pixel masks called “label images” (see Table S1). In general, the label format depends on which model one opts to use. Some label creation tools automatically transform masks into labels; our workflow additionally provides scripts to convert masks into labels.

4.5.3 Training

Once users have prepared image tiles, masks, and labels, they can proceed with training. Generally, this stage involves showing an AI/ML model image tiles, having it make predictions, and

then comparing those predictions to corresponding labels (i.e., the “ground-truth”). By repeating this process, a model gradually “learns” how to identify features of interest (Goodfellow et al., 2016). Following training, a model can take a new tile as input and produce its own labels, such as predicted bboxes or pixel-level masks, that represent where the features of interest are and, if applicable, what shape they have (e.g., Goodfellow et al., 2016; Zhu et al., 2017).

4.5.3.1 Model Choice Our workflow divides the task of identifying features into two main steps: object detection and instance segmentation. Detection locates each object in the image, whereas segmentation outlines the exact shapes of the objects (e.g., Everingham et al., 2010; He et al., 2017; Zhu et al., 2017). We separate segmentation from detection so that users can perform either task independently (e.g., run segmentation without first utilizing or retraining a detection model). Importantly, our workflow is agnostic to the choice of model for either task, and users may opt to deploy different models than the ones we use.

4.5.3.1.1 Object Detection Detection models are valuable for tasks such as tracking the count of objects or identifying positional changes over time. In our workflow, the detection model takes image tiles and returns bboxes around features of interest in those tiles, each with a confidence score. We ignore detections that have a low confidence value (e.g., < 0.25) and merge bboxes that overlap across tiles. Like for the input labels, the corner coordinates of these bbox outputs are stored in text files. Our detection model of choice is YOLOv8 due to its speed and consistency (Varghese & Sambath, 2024).

4.5.3.1.2 Instance Segmentation Users are often interested in the exact shape of features of interest (e.g., a tight boundary around agricultural infrastructure), which, by themselves, detection models do not directly provide. To generate such information, we leverage an instance segmentation model in our workflow to outline individual objects (i.e., instances) and output a pixel-level mask for each instance. We use the Segment Anything Model (SAM), which is a foundation model (i.e., one that has been trained on extensive, general-purpose datasets; (Bommasani, 2021)) that can

perform segmentation from bboxes or points (e.g., the center point of an object) and is known to work effectively on nadir-view imagery (Albughdadi et al., 2024; Kirillov et al., 2023; Lu et al., 2025; Osco et al., 2023). Specifically, we feed bbox labels from our detection model into SAM, which then outputs a corresponding instance mask. While additional training using customized image-label data might improve SAM segmentation performance, we opt not to perform such training in our implementation.

4.5.3.2 Expert-in-the-Loop Refinement Since our goal is to calibrate model uncertainty for downstream tasks, we include expert validation routines in our workflow. Such validation can occur during training (i.e., for fine-tuning a model) and/or after inference (i.e., on outputs produced by a fully trained model), with the former approach ensuring that a user can repeatedly refine a model. The structured evaluation framework we have developed relies on custom graphical user interfaces (GUIs) to read and visualize outputs (i.e., tiles) from detection and segmentation models (Figure 4.2). These GUIs enable experts to mark FP, find FN, and verify TP. Through these GUIs, we collect feedback and, optionally, update training data to improve model performance. Additionally, the validation data also serve as the backbone of the calibrated uncertainties our workflow produces (Section 4.5.4.1). Critically, users do not need a background in AI/ML to run these GUIs or provide usable feedback. As a result, subject matter experts can participate in model development and evaluation.

In the case of FPs, our “FP Checker” GUI allows experts to examine mask predictions output by a model (Figure 4.2a). FP Checker displays a cropped image tile with a predicted instance location (marked by a red triangle), prompting the expert to confirm whether this detection is true or false. The expert is further able to provide notes to flag concerns about the instance in the review (e.g., noting cases where two instances are recognized as one). Experts can choose to evaluate the full image set, some number of random images, or a subset based on specific criteria (e.g., those with a SAM mask score < 0.95).

Identifying FNs is harder than reviewing FPs because missed detections do not appear as masks

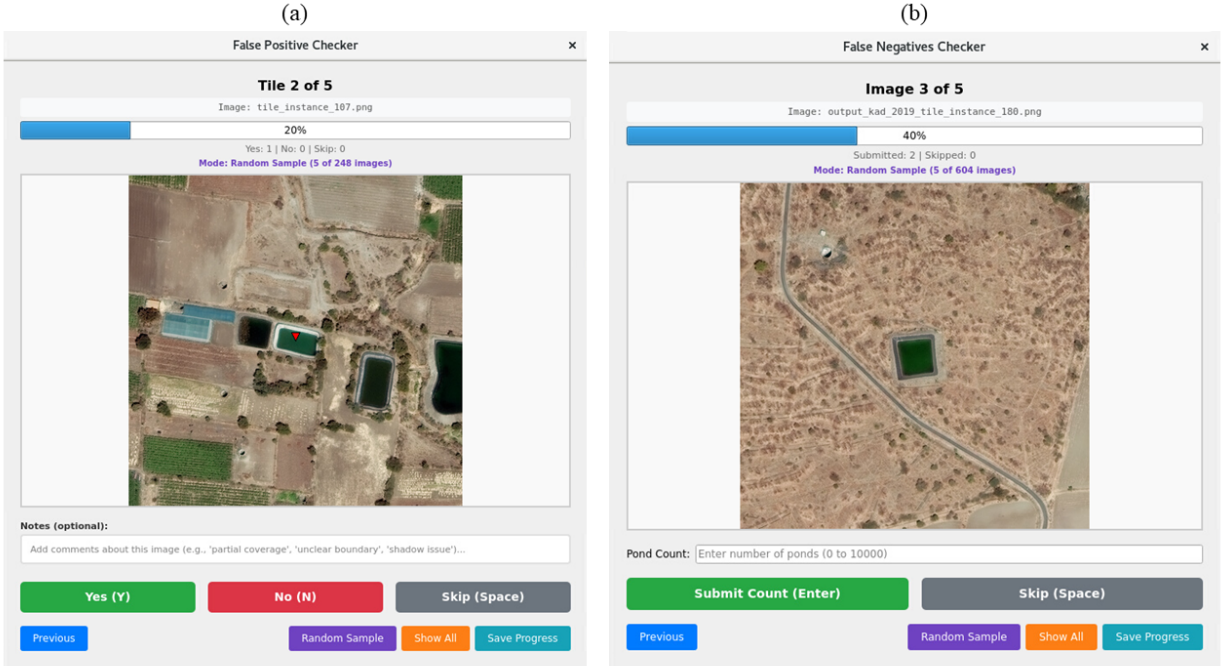


Figure 4.2: **Expert Verification GUIs.** (a) False Positive (FP) Checker and (b) False Negative (FN) Checker. A progress bar shows how many tiles the expert has reviewed or skipped.

in outputs. As a result, our “FN Checker” GUI allows experts to inspect imagery to find FNs (Figure 4.2b). The FN Checker presents randomly cropped image tiles and asks the expert to enter the number of instances that they see in the tile. We then compare the user’s count of instances with the model’s predicted count to identify missed detections. Like in the FP checker, the expert can choose to evaluate all or a subset of the input satellite imagery.

After each training round, we run the FP and FN Checkers once on the predictions. We document expert inputs in a comma-separated values (.CSV) file along with the tile name, its upper-left corner coordinates, the review outcome, and optional notes to provide context. We update the training set by replacing tiles with incorrect masks using the same tiles that have corrected, expert-traced masks, and we merge these corrected masks with the original hand-traced masks to form a verified ground-truth dataset. We then use the corrected masks to fine-tune our detection or segmentation models.

4.5.4 Post-processing

After the final training round, we run the FP and FN Checkers to verify the output. Then, we merge all masks from each tile into a single, labeled image (which we georeference, if necessary). These data, along with the trained detection and segmentation models and calibrated uncertainties, serve as the central outputs of our workflow (Figure 4.1).

4.5.4.1 Evaluation Metrics Following the completion of model training, we generate evaluation metrics that end-users can readily leverage in their work. This *translational* step involves us first identifying metrics that are useful to the task at hand and then calculating errors/uncertainties by comparing outputs from the trained models to ground truths (incorporating, wherever possible, data from the expert feedback in Section 4.5.3.2). The resulting calibrated metrics are ready for use in downstream tasks. For example, if an accurate measurement of area is valuable for a user, a report of how close (or far) a model’s prediction of area is to that of an expert’s (i.e., *areal uncertainty*) might serve as an ideal and actionable calibrated metric.

4.6 Case Study

4.6.1 Lined Farm Ponds in Maharashtra, India

In the western Indian state of Maharashtra, the past decade has seen the proliferation of a peculiar adaptation known as the “lined farm pond” (LFP). Farmers construct these ponds as raised storage pools, which they then line with a high-micron plastic sheet (Kale, 2017; Shah et al., 2021; Sidhu & Shah, 2021). In many drought-prone areas, farmers can pump out groundwater and store it above-ground in LFPs to gain private control over limited water resources (Prasad & Sohoni, 2018). As a result, LFPs serve the opposite purpose of conventional farm ponds, which capture precipitation and overland runoff for the purpose of recharging groundwater sources. For various reasons, the expansion and intensification of cultivation in drought-prone areas, assisted by LFPs, may be considered a maladaptation. First, farmers who engage in such agro-ecological decisions

may be rendered more vulnerable to climate shocks and variabilities. Second, LFPs can create inequality based on who can pump and store the water resources (Prasad & Sohoni, 2018). LFPs can also lose substantial amounts of water to evaporation because they are uncovered, the water stored within them is readily evaporated, leading to substantial village system water losses (Sidhu & Shah, 2021). Understanding the scale, location, and impact of land and water-use change problems is critical for reducing long-term environmental and livelihood risks to communities.

A long-term, village-scale, accounting of LFP proliferation would enable researchers and stakeholders to assess land- and water-use change and potential maladaptive impacts resulting from these behaviors. To date, there are no such publicly available long-term records, in part because it is labor-intensive to manually map LFPs. Researchers who evaluated the use of convolutional neural networks (CNNs) for LFP detection (Bera et al., 2025; Tundia et al., 2023) have acknowledged the need for improved training datasets because currently available datasets are limited in time and image resolution, making them insufficient for long-term and multi-village monitoring. Furthermore, earlier studies do not provide application-specific outputs, such as the areal extent and shape of LFPs. In our case, we integrate detection, segmentation, expert correction, and quantification in a unified framework. This workflow fills existing knowledge gaps and produces analysis-ready outputs.

4.6.2 Data

We use two high-resolution (i.e., $< 60\text{cm/pixel}$) satellite images (Table S2) to quantify the change in number and area of LFPs in Kadwanchi, which is a village in Maharashtra, India that covers 1,607 hectares (Pawar et al., 2024). These images—one from 2011 and the other from 2019—came as 16-bit, 4-band Tagged Image File Format (.TIFFs) files with Universal Transverse Mercator (UTM) coordinates referenced to the World Geodetic System 1984 (WGS 84) datum.

4.6.3 Workflow application

We began by manually tracing ~ 100 LFPs on the full satellite image. Next, we turned these outlines into masks and converted them to YOLO format text labels. We then padded and tiled the full satellite image and its corresponding LFP masks into 1024×1024 pixel patches with a stride $s = 256$, and then removed background tiles using a threshold. For the Kadwanchi 2019 image, tiling after filtering yielded approximately $T \approx 530$ tiles for training. We held out 20% of those tiles as a validation dataset.

Following our workflow, we trained the detection model on the tiled image–label pairs from the Kadwanchi 2019 image only. We fine-tuned a YOLOv8 detection model initialized from the official pretrained YOLOv8 model on a single NVIDIA RTX A6000. We performed 10 independent fine-tuning runs (100 epochs each; one full pass over the training dataset per epoch). We used the AdamW optimizer with a learning rate of 0.01 (see Table S1). We then selected the model with the best performance when applied to the validation tiles (specifically, we chose the model with the highest mAP; see Table S1) and applied it to tiles of the satellite image to produce bboxes. Then, for segmentation, without additional model training, we passed each tile that had bbox text labels to SAM as a prompt for mask generation for each tile.

We provided every detection and a randomly selected subset of tiles (10% of the total) to an expert for review. Using the FP and FN Checker GUIs, this expert flagged both mislabeled and missed LFPs. This process was time-efficient: reviewing a random sample of five tiles took an average of under 30 seconds for both FPs and FNs. We converted the expert’s flags into YOLO labels and added these data to our training set, which we used to fine-tune our original model.

To train the models with the updated dataset after expert review—which we refer to as trained iteration M_1 (M_0 being the original)—we performed a single run of 100 epochs and employed the AdamW optimizer with a smaller learning rate of 0.001. Next, we applied M_1 to both the 2011 and 2019 imagery and then had experts validate the outputs of that trained iteration using the FP and FN Checkers. Finally, we calculated a set of calibrated uncertainties.

4.6.4 Uncertainty Calibration

For both images, we matched the output from M_1 with ground truth annotations and computed a set of performance metrics. In dialogue with experts, we determined that two metrics—counting the number and amount of occupied area within a village—directly align with the task of tracking LFP proliferation over time.

With respect to the count of LFPs:

$$\text{Predicted Count} = \text{TP} + \text{FP}$$

$$\text{True Count} = \text{TP} + \text{FN}$$

and

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1-score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We define the *absolute count error* (ACE) as the difference between the predicted count and the true count:

$$(\text{TP} + \text{FP}) - (\text{TP} + \text{FN}) = \text{FP} - \text{FN}$$

To convert absolute count error into a relative uncertainty that we can use as a reference when applying M_1 to new imagery, we define *relative count error* (RCE) as:

$$\frac{\text{FP} - \text{FN}}{\text{TP} + \text{FN}} \times 100\%$$

To calculate areal uncertainty for an instance j , we compare the predicted area, A_j^{pred} , to the expert-labeled area, A_j^{gt} . To assess the areal uncertainty across an entire village with n instances, we compute the total relative error (RAE):

$$\frac{\sum_{j=1}^n A_j^{\text{pred}} - \sum_{j=1}^n A_j^{\text{gt}}}{\sum_{j=1}^n A_j^{\text{gt}}}.$$

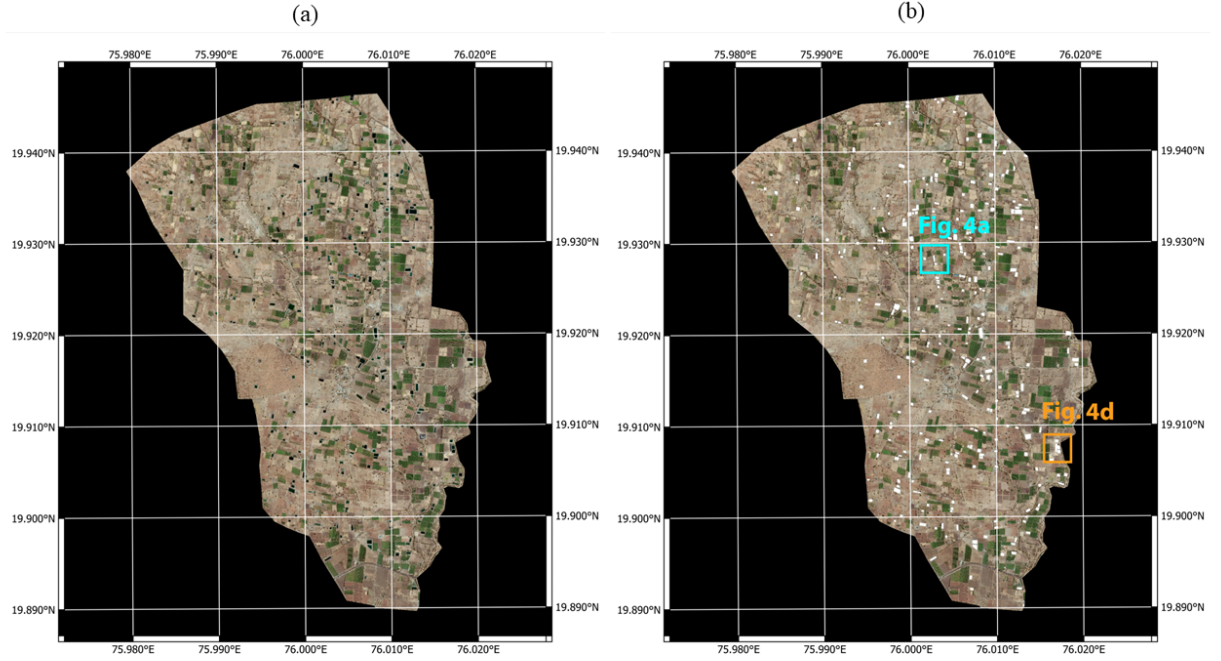


Figure 4.3: **LFP detection applied to imagery of Kadwanchi village.** (a) Raw 2019 satellite image and (b) predicted LFP masks on the village map (in white) from M_1 . Highlighted boxes on the map indicate the tiles shown in Figure 4.4a–c (cyan) and Figure 4.4d–e (orange).

Note that we have defined our relative areas so that they reflect a model’s bias: positive RCE/RAE values mean the model overestimates counts/area, while negative RCE/RAE values indicate that the model underestimates counts/area.

4.7 Results and Discussion

4.7.1 Results

Our application is able to detect LFPs of varying sizes and shapes across satellite imagery (Figure 4.3). The resulting maps of instances are ready for various downstream tasks, including tracking the change in the number of LFPs in a given area over time (provided there is imagery available for evaluation).

By using an expert-validated training set, we were able to increase detection performance (Table 4.1), with M_1 detecting more LFPs ($n = 280$) than M_0 ($n = 248$; Figure 4.4a-c). This

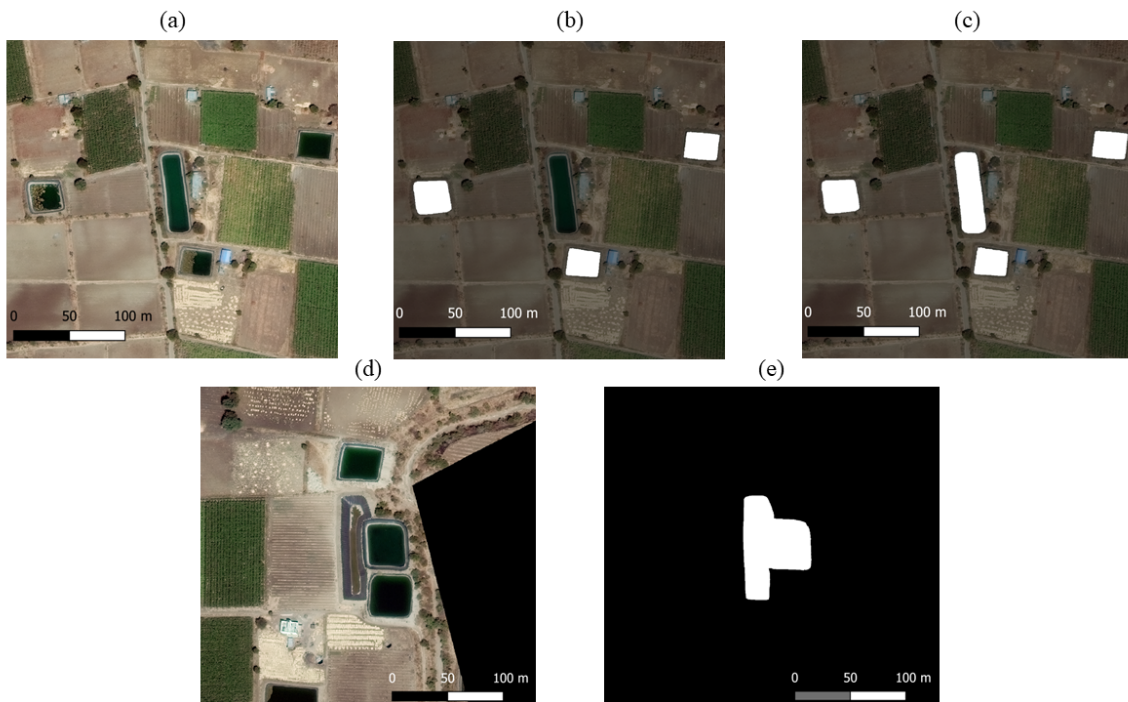


Figure 4.4: **Expert-validated model outputs on 2019 Kadwanchi imagery.** (a) Original image tile. (b) Prediction from M_0 . (c) Prediction after expert verification and retraining (M_1). (d) Tile of the FN example, (e) Mask prediction of the FN example from (d). In this case, two closely spaced LFPs are recognized as a single entity, causing the unusually shaped LFP to be only partially detected by the model.

improvement was especially pronounced when we applied the models to the older, lower-resolution, 2011 imagery: M_0 found only 25 LFPs while M_1 detected 53 (out of a known total of 60). Driving this increased effectiveness was a reduced number of FNs on both sets of imagery.

When looking at downstream task metrics (Table 4.1), M_1 provided better estimates of both LFP area and count. As above, this improvement was also significant in older imagery, where the area error went from -51.9% to -7.3% .

Importantly, one iterative round is not enough to produce perfect outputs. For example, M_1 cannot recognize the distinct instances when LFPs are near one another (i.e., Figures 4.4d and 4.4e). In such cases, the SAM mask score for the FN (e.g., $m = 0.8177$) may be below the average SAM mask score for all LFP detections (e.g., $m = 0.95$). By filtering on mask scores, experts can identify edge cases and update the training set for future iterations.

Table 4.1: Detection performance and downstream task metrics on Kadwanchi images for two trained iterations.

Year	Trained iteration	Detection performance					Downstream task metrics					
		Detections	TP	FP	FN	Precision	Recall	F1	Total area (ha)	RAE (%)	ACE	RCE (%)
2019	M_0	248	248	0	33	1.000	0.883	0.937	31.72	-14.1	-33	-11.7
2019	M_1	280	280	0	1	1.000	0.996	0.998	36.79	-0.3	-1	-0.36
2011	M_0	25	25	0	35	1.000	0.417	0.588	4.12	-51.9	-35	-58.3
2011	M_1	53	53	0	7	1.000	0.883	0.938	7.95	-7.3	-7	-11.7

^aTotal LFP area was converted from pixel counts using the year-specific spatial resolution in Supplementary Table S2.

For 2011 WorldView-2 imagery, 1 pixel = 0.25 m² (50 cm/pixel); for 2019 WorldView-3 imagery,

1 pixel = 0.09 m² (30 cm/pixel). Ground-truth LFP areas were 8.57 ha in 2011 and 36.92 ha in 2019.

4.7.2 Discussion

Our case study results demonstrate that our workflow can deliver high-quality, analysis-ready outputs along with calibrated performance metrics that can be easily understood by users unfamiliar with the technical details of computer vision models. Critically, none of the techniques or metrics that our proposed method leverages are, by themselves, novel. Rather, the structured framework that we present enables repeatable and actionable validation of remotely sensed object detection. Such a workflow shifts the focus from designing new model architectures to ensuring existing detection and segmentation models can inform real-world scenarios, such that researchers may focus on applications rather than technical optimizations.

Expert input and iterative updates may keep an AI/ML application’s performance consistent as it encounters previously unseen regions or sensor configurations, which researchers may have to contend with in studies that cover long time periods. Multiple expert-validated training iterations can mitigate the gap between training data and unseen data conditions as more challenging cases are reviewed, corrected, and incorporated into the subsequent training dataset. For example, M_0 achieved high detection rates for square-shaped LFPs but struggled with elongate ones. Since the latter represented a pattern that was not common in the M_0 ’s training data, we were able to address the issue by including such LFPs in the training set for the updated M_1 .

A limitation of our framework is the labor required for expert review, particularly when applied over larger regions or datasets. However, our workflow is intended to support iterative validation rather than an exhaustive manual correction of all outputs at once. In larger-scale applications (e.g, thousands of image tiles), expert effort can be focused through random or stratified sampling, review of error-prone cases, and task-specific stopping criteria based on the needs of the end use. As additional verified samples are incorporated into training rounds, the need for intensive review may decrease, although the appropriate level of review will still depend on the downstream task and the level of accuracy required by users.

This work provides an example of what it means to build more “confident” AI/ML applications. Through uncertainty calibration (both by creating task-specific metrics and estimating performance

via expert validation of model outputs on unseen data), we provide a way for end-users to easily estimate and communicate confidence in an AI/ML application’s effectiveness. Critically, while expert involvement in the validation of predictive AI/ML applications may contribute to increasing their “trustworthiness” for end-users (McGovern et al., 2024), we refrain from making larger assertions about our workflow’s trustworthiness because trust is a social and relational concept that involves multiple dimensions of use, privacy, and accuracy (see Bostrom et al., 2024).

4.8 Conclusion

We have presented a modular workflow that (i) streamlines the training and deployment of object detection and segmentation models, (ii) leverages expert guidance to improve detection and segmentation performance and generate calibrated uncertainties, and (iii) provides task-based evaluation metrics that are understandable and actionable by end-users. The Kadwanchi case study illustrates the power of our expert-in-the-loop approach and how calibrating uncertainty can turn AI/ML application outputs into metrics available for decision-making and policy applications. In totality, our workflow offers a template for a standardized expert-in-the-loop framework that can support other geospatial detection and segmentation tasks when paired with application-specific review criteria and uncertainty measures.

Looking beyond the technical realm, we believe that our case study raises important concerns about the development, deployment, and use of AI/ML applications, all of which will require careful consideration. For instance, it is not unlikely that our approach—which can provide quantitative measures of change that could implicate policy- and decision-makers at multiple scales—could be used to surveil and punish farmers. Future research, therefore, must consider ethical and responsible frameworks for employing AI/ML applications.

4.9 Open Research

We plan to release all preprocessing, detection, segmentation, and post-processing code as a plug-and-play Python package (ponds).¹ We will publish the code on GitHub and Dryad after acceptance.²

The repository includes the package, example workflows, and configuration files (e.g., `config.yml`, `environment.yml`) to reproduce our results or adapt our workflow to other datasets.

¹<https://github.com/reconstructionLab/ponds>

²<https://tinyurl.com/3z2ccb2a> (Link for reviewers)

Table S1: Glossary (methods-related technical terms)

Term	Definition
Bounding box	A rectangle that encloses the object users are trying to detect in an image (Everingham et al., 2010).
Instance mask (mask)	The exact shape (in pixels) of one object in the image; used to measure area and track change over time (He et al., 2017).
Instance segmentation	A method that detects each object and outputs its exact shape (He et al., 2017).
Intersection over Union (IoU)	A score (0–1) measuring overlap between the model’s predicted region and the human-traced region; higher is better (Everingham et al., 2010).
Label	Human-traced boxes or shapes that mark object locations, used for training and evaluation (Goodfellow et al., 2016).
Label image	An image where each pixel stores a numerical label indicating which object it belongs to. For example, with background pixels typically set to 0, pixels of pond 1 take the value 1, pixels of pond 2 take the value 2, and so on.
Learning rate	Learning rate refers to the size of the steps that the algorithm takes when optimizing models (Goodfellow et al., 2016).
Mean Average Precision (mAP)	A score (0–1) that rewards models finding the correct objects and avoiding mistakes, averaged across easier to stricter overlap requirements (see IoU) (Everingham et al., 2010).
Object detection	A method that locates each object and draws a bounding box around it (Everingham et al., 2010).
Optimizer	The method used during training to adjust the model step by step so its predictions better match the data. (Goodfellow et al., 2016).
Tiling	Cutting a large image into smaller pieces.

Table S2: Specifications of satellite imagery datasets^a

Site (Date)	Satellite	Resolution (cm/pixel)	Bands (nm)
Kadwanchi (2011-12-25)	WorldView-2	50	Blue 450–510 Green 510–580 Red 630–690 Near-infrared 770–895
Kadwanchi (2019-03-01)	WorldView-3	30	Blue 445–517 Green 507–586 Red 626–696 Near-infrared 765–899

^aBand specifications from Maxar documentation (Maxar Technologies, 2020a, 2020b).

5 Chapter 3: Less is More? Evaluating Geospatial Foundation Models for Soil Geochemistry Prediction Under Data Scarcity and Geographic Shift

5.1 Relevance

Chapter 3 examines the reliability of geospatial foundation models (Geo-FMs), which can possibly predict soil properties in regions where field sampling is sparse. I benchmark frozen Geo-FM embeddings for predicting the major soil elements Al, Ca, Na, and K across the conterminous United States, testing whether prediction accuracy holds across years, sampling densities, and held-out regions. The embeddings improve over raw spectral features and remain stable across imagery years, but their accuracy still depends on training-site density, coverage in the embedding space, and regional geology. Guidance is built on environmental context, since satellite imagery alone does not contain all the information the prediction requires. Adding surficial geology and soil survey covariates reduces error in the held-out region, although the gain varies by element. These results show that reliable Geo-FM prediction in data-sparse regions depends on sampling design and geological context as much as on model scale.

5.2 Status

This work has been submitted to the 34th ACM International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2026) and is currently under review. Lin, P.-C., & Mehra, A. (under review). Less is More? Evaluating Geospatial Foundation Models for Soil Geochemistry Prediction Under Data Scarcity and Geographic Shift. *ACM SIGSPATIAL 2026*.

5.3 Chapter Abstract

Measurements of elemental geochemistry, which provide insights into soil fertility, nutrient availability, and ecosystem health, are spatially limited due to sparse field sampling. Recent efforts leveraging geospatial foundation models (Geo-FMs) have shown promise for addressing this limitation. Even so, it remains unclear whether the embedding architecture of Geo-FMs enables them to reliably predict properties when training labels are sparse, unevenly distributed, or missing from target regions, which is necessary to produce globally-extensive estimates of soil geochemistry. Here, we benchmark frozen OlmoEarth embeddings using 4,814 sites across the conterminous United States, targeting four elements: Aluminum, Calcium, Sodium, and Potassium. Our OlmoEarth-embedding-derived model exhibits improved performance over one trained on raw Sentinel-2 spectral features, reaches R^2 values up to 0.565, and remains stable across 2020–2025 imagery. Training-sample experiments show that embedding proximity can identify which measured sites are most useful for a target region, while the late fusion inclusion of co-variates, including geology, improves predictions for all four elements. These results suggest that Geo-FM embeddings can help predict soil geochemistry with limited labels, but are most effective when combined with careful sample selection and additional co-variates (e.g., geology).

5.4 Introduction

The elemental composition of soil, which is a reflection of its parent material, weathering, and landscape history, determines its fertility, agricultural productivity, and capacity for ecosystem services (Dominati et al., 2010; Jenny, 1994). Digital soil mapping (DSM) of elemental composition necessitates labor-intensive field sampling campaigns and laboratory analyses, making it challenging to achieve broad spatial coverage. For example, in the conterminous United States (CONUS), large-scale national surveys cover just a fraction of the landscape, have samples concentrated in accessible areas, and under-represent geologically distinct or remote regions (Hengl et al., 2021; Minasny & McBratney, 2016; Smith et al., 2013).

Since the reflectance of soils is, in part, a function of mineral composition, satellite imagery can help identify their composition (e.g., Mulder et al., 2011). Indeed, geospatial foundation models (Geo-FMs) pretrained on large satellite imagery datasets have shown the potential to predict various properties of soils. For example, FoundationSoil utilizes frozen Prithvi-EO-2.0 embeddings to generate estimates of organic carbon across CONUS (R^2 up to 0.72) (Song et al., 2025). To date, however, it is unclear whether Geo-FM representations can predict soil elemental compositions and, more importantly, whether outputs remain reliable when direct soil measurements are sparse or unavailable.

To address this knowledge gap and build the foundation for an accurate and globally-extensive estimate of soil geochemistry, we test the ability of Geo-FM embeddings to predict four elements Aluminum (Al), Calcium (Ca), Sodium (Na), and Potassium (K). To evaluate Geo-FM-based soil geochemical prediction across the CONUS, we design experiments around five key questions: (1) Do models leveraging pretrained Geo-FM embeddings improve on those that utilize spectral baselines? (2) Are Geo-FM embeddings temporally stable across different years? (3) Does sampling density affect prediction accuracy, and how few labeled sites does a Geo-FM need before accuracy plateaus? (4) Does embedding proximity predict which training sites are most useful for held-out regions in the training data? (5) When a Geo-FM fails to generalize to an out-of-distribution (OOD) region, can incorporating surficial geology and soil survey co-variates via late fusion improve predictive accuracy?

Ultimately, we:

1. Present a benchmark of Geo-FM embeddings for predicting Al, Ca, Na, and K across CONUS, evaluating model scale, temporal consistency, sample efficiency, and geographic generalization.
2. Demonstrate that Geo-FM soil element predictions drop in held-out regions, but selecting similar training sites in the embedding space can improve accuracy, illustrating how sampling design can support prediction in OOD regions.

3. Illustrate that Geo-FM embeddings with late fusion geology and soil co-variates improve elemental prediction (with an R^2 of up to 0.62).

5.5 Methodology

5.5.1 Data

We use the United States Geological Survey (USGS) DS-801 soil survey database, which contains field-measured element concentrations for 4,814 sites (each corresponding to a point coordinate) across CONUS (Smith et al., 2013). We first extract the 0–5 centimeter (cm) horizon means of Al, Ca, Na, and K (reported in weight percent, *wt%* of dry soil). We then generate 64×64 pixel windows of Sentinel-2 (S2) Level-2A imagery (10 meter (m) resolution) centered on each site’s point coordinate. We assemble twelve monthly composites from January to December 2020, representing each site as an input tensor with shape (64, 64, 12, 12), corresponding to window length, window width, spectral bands, and months. We filter out sites that lack a complete 12-month S2 tensor. We use the S2 image tensors to create Geo-FM embeddings for each soil-sampling site (see Section 5.5.3, and Figure 5.1 for a visualization of a training site). To test the stability of embeddings over multiple years, we repeat the same S2 extraction procedure on same site coordinates separately for each year from 2020 to 2025 (see Section 5.5.5).

5.5.2 Geology and Soil Co-variates

As part of our experiments, we evaluate two co-variates:

5.5.2.1 DS-425 Surficial Geology (74 dimensions) From the USGS DS-425 national surficial geology map, we extract feature vector for every site that includes: Unit Codes, rock types (carbonate, residual, glacial, volcanic), unit thickness, unit name classification (11 classes), texture groups, and geological age (Soller et al., 2009). For Unit Codes, we use a 50-dimensional encoding that keeps the 49 most common codes as separate categories and all remaining or missing codes into one rare/missing category.

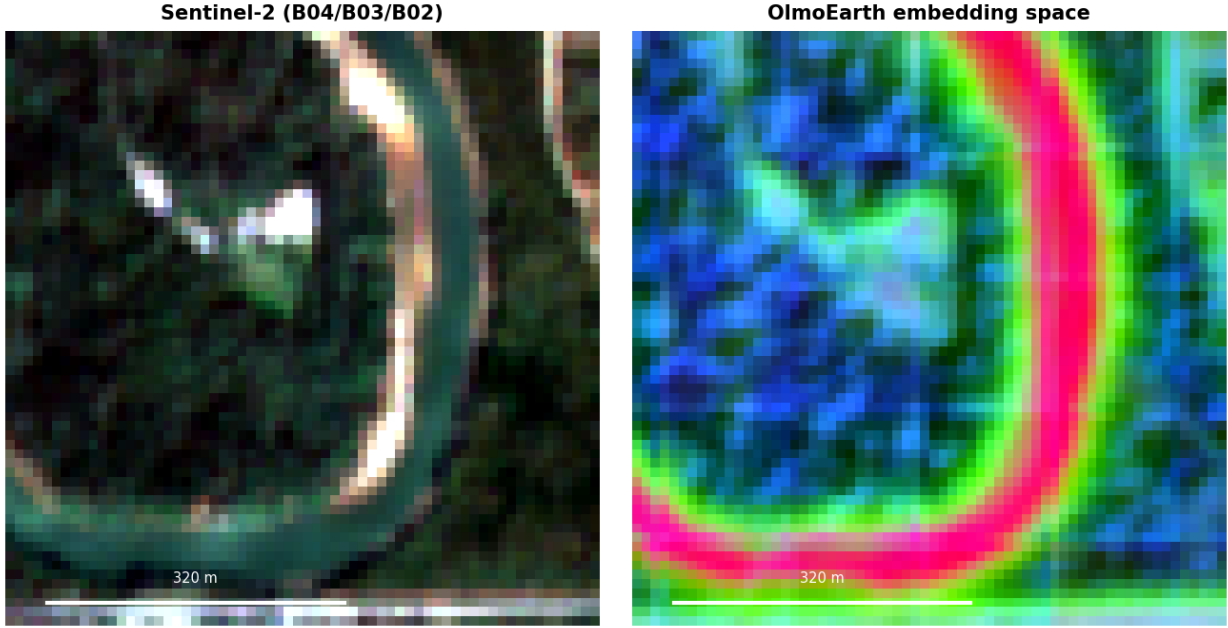


Figure 5.1: Training site sample image. Left: S2 true-color image. Right: PCA false color of per-pixel frozen OlmoEarth embeddings (PC1–PC3).

5.5.2.2 Soil Survey Geographic Database (SSURGO) Soil Survey (38 dimensions) From a pre-extracted 19-band SSURGO raster at 10 m resolution, we compute per-band spatial mean and standard deviation across the 64×64 pixel window centered on each site. The 19 bands include physical properties of soil at 0–30 cm and 0–100 cm horizons, such as sand, clay, pH, landscape properties such as slope, and three categorical soil attributes (drainage class, hydrologic group, and taxonomic order).

5.5.3 Geo-FMs

Geo-FMs pretrained on large satellite datasets produce embeddings that researchers can transfer to downstream tasks. Recent examples of Geo-FMs include Prithvi-EO-2.0, AlphaEarth, and OlmoEarth (Brown et al., 2025; Herzog et al., 2025; Szwarcman et al., 2025). We use OlmoEarth, pretrained on S2 imagery, because its released weights make it suitable for reproducible frozen-embedding benchmark experiments and fine-tuning (Herzog et al., 2025). We evaluate four model weights of different dimensions from OlmoEarth: Nano (128 dimensions), Tiny (192 dimensions),

Base (768 dimensions), and Large (1024 dimensions). All backbone weights are frozen throughout downstream training. For each DS-801 site, following the process to generate embeddings from OlmoEarth, we pass the corresponding 12-month S2 through the backbone, then mean-pool all valid tokens to produce a single D -dimensional embedding vector (Herzog et al., 2025). For the purposes of comparison, each site is also represented by a 144-dimensional image-only baseline, computed by averaging the 64×64 patch per S2 band and per month and flattening across all 12 bands and 12 months.

5.5.4 Prediction Head

All downstream soil chemistry models use a lightweight three-layer ReLU MLP head on the frozen embeddings, in the spirit of (Song et al., 2025): the input feature dimension D (144 for the baseline) maps to 128 hidden units (ReLU, 0.1 dropout), then to 64 hidden units (ReLU), then to a single scalar prediction. We train a separate head for each element. We standardize all targets using the mean and standard deviation of the training set, then convert back to original units for evaluation. We train each model with AdamW using a learning rate of 1×10^{-4} , weight decay of 1×10^{-4} , batch size 256, and up to 30 epochs, stopping after five epochs without validation-loss improvement.

5.5.5 Experiment Protocols

5.5.5.1 Experiment 1: Geo-FM soil prediction performance We evaluate whether predictions from FM embeddings improve over raw S2 bands and model size affects accuracy using both R^2 and RMSE as performance metrics. We use a 10-fold spatial cross-validation (CV) to mitigate spatial autocorrelation. This process involves assigning the DS-801 sites to $200 \text{ km} \times 200 \text{ km}$ grid cells, shuffling the cells, and then dividing them into 10 equal groups. In each fold, one group is the test set, and the remaining nine groups form the training set, including one validation group for early stopping ($\sim 80\% / 10\% / 10\%$). We report the mean and standard deviation for R^2 and RMSE across the 10 test folds (see Table 5.1).

5.5.5.2 Experiment 2: Temporal stability We explore whether embeddings remain consistent over multiple years by testing whether a location’s embeddings from different annual S2 composites are close to one another. We extract annual embeddings between 2020 and 2025 for each location after dropping out missing data to $n = 4,801$ in the DS-801 database, measuring within-site temporal drift as the L_2 distance and cosine similarity between embeddings from consecutive years ($n_{total} = 24,005$). For comparison, we compute the same metrics for 200,000 randomly sampled pairs of distinct sites from 2020 embeddings. If embeddings are temporally stable, within-site year-to-year distances should be much smaller (and cosine similarities much higher) than distances between different sites.

5.5.5.3 Experiment 3: Training sample density We examine how many labeled training sites are necessary before R^2 plateaus, and whether performance collapses at low sampling density. Reusing the same fixed 10-fold spatial CV split as the first two experiments, we sub-sample the training pool by drawing sites uniformly at random without spatial stratification. To ensure that a single random choice of training sites or model initialization does not solely drive the results, we repeat training at each “density” level three times using different random seeds.

5.5.5.4 Experiment 4: Regional holdout and learning from feature space proximity We test whether proximity in embedding feature space predicts how well a model can predict “nearby” sites.

We apply k-means ($k=15$) to the L_2 -normalized OlmoEarth Base model embeddings of all sites and select the focal cluster, F with the most samples. For each non-focal site, i , where e_i denotes the embedding of site i , we define its distance to F as

$$d(i, F) = 1 - \frac{1}{|F|} \sum_{j \in F} \tilde{e}_i^\top \tilde{e}_j,$$

where $\tilde{e} = e / \|e\|_2$, and j indexes the sites in F . We rank non-focal sites by increasing $d(i, F)$, where smaller values indicate greater embedding proximity. Within the non-focal sites, we divide the

nearest 15% in distance to the focal sites into three 5-percentile test zones to test how embedding space proximity affects model performance: Zone A (1–5%), Zone B (6–10%), and Zone C (11–15%). These zones serve as the held-out test sets that represent different proximity to the focal sites, while F is the treatment variable.

We compare two training sets, each with $N = 2,000$ sites: INCL and EXCL. The former contains F , while EXCL replaces F with the same number of non-focal sites; the remaining background sites are shared. For each target element and run, we compute

$$\Delta R_z^2 = R_{\text{INCL},z}^2 - R_{\text{EXCL},z}^2$$

where $z \in \{A, B, C\}$ denotes the embedding-proximity zones. Positive ΔR_z^2 means that including F improves prediction in that zone beyond the effect of training-set size.

5.5.5.5 Experiment 5: Out-of-distribution holdout and co-variate mitigation Finally, we use late fusion to add mapped geologic and soil-survey co-variates (see Section 5.5.2) to frozen Geo-FM embeddings via concatenation. To understand how this approach may work in geographic out-of-distribution settings (i.e., to simulate deployment to a region with no local labels), we also hold out all DS-801 sites in FL, GA, SC, and NC. Across three runs, we compare the prediction performance of frozen Base embeddings against our late fusion vectors by evaluating holdout performance with R^2 , RMSE, and per-site absolute error.

5.6 Results and Discussion

5.6.1 Results

For Experiment 1, all embedding-based predictions outperform the image-only baseline across all four targets (Table 5.1), with performance increasing from Nano to Tiny to Base. Results with Large are mixed: Ca predictions are slightly improved ($R^2 = 0.347$ versus 0.342), but performance on Al, Na, and K is worse.

Table 5.1: Soil element prediction performance. Values are the mean with standard deviation in parentheses. Bold values indicate the best result for each element.

(a) R^2 performance				
Model / Features	Al	Ca	Na	K
S2 Image-only	0.307 (0.099)	0.175 (0.074)	0.376 (0.092)	0.325 (0.112)
OlmoEarth Nano	0.384 (0.109)	0.195 (0.062)	0.413 (0.078)	0.374 (0.122)
OlmoEarth Tiny	0.447 (0.109)	0.221 (0.075)	0.447 (0.078)	0.402 (0.120)
OlmoEarth Base	0.565 (0.091)	0.342 (0.097)	0.551 (0.074)	0.478 (0.108)
OlmoEarth Large	0.549 (0.098)	0.347 (0.106)	0.542 (0.074)	0.469 (0.107)
(b) RMSE performance (wt%)				
Model / Features	Al	Ca	Na	K
S2 Image-only	1.733 (0.130)	2.364 (0.510)	0.523 (0.052)	0.588 (0.045)
OlmoEarth Nano	1.633 (0.155)	2.335 (0.491)	0.510 (0.066)	0.565 (0.045)
OlmoEarth Tiny	1.544 (0.144)	2.295 (0.482)	0.495 (0.069)	0.552 (0.045)
OlmoEarth Base	1.366 (0.104)	2.093 (0.374)	0.446 (0.065)	0.516 (0.042)
OlmoEarth Large	1.391 (0.114)	2.079 (0.359)	0.450 (0.065)	0.520 (0.043)

Experiment 2 reveals that same-site embeddings across consecutive years have a lower mean L_2 distance and higher mean cosine similarity value than random different-site pairs (1.89 vs. 4.99 and 0.97 vs. 0.80, respectively; Figure 5.2).

In Experiment 3, performance increases most sharply between 500 and 1,000 training sites, showing that Geo-FM-based soil prediction is sensitive to labeled-site density at low sample sizes (Figure 5.3). Al and Na approach a plateau near $R^2 \approx 0.55$ after $n = 1,000$, while K continues to improve more gradually, reaching $R^2 = 0.478$ at the full training pool. Ca predictions perform poorly throughout (max $R^2 = 0.342$).

For Experiment 4, the focal cluster we select ($n = 541$; Figure 5.4) largely comprises sites in the eastern and southeastern US. The benefit of including this cluster in training is concentrated in the embedding-nearest test sites, with the largest performance gains occurring in Zone A (the nearest 1–5% of non-focal sites) for all four elements (Figure 5.5).

Experiment 5 shows that adding co-variates reduces errors and improves performance, increasing R^2 to 0.624 for Al, 0.419 for Ca, 0.604 for Na, and 0.506 for K under the same CONUS spatial CV protocol (Table 5.2). This improvement also can be seen in holdout data (Table 5.3) and is especially pronounced for Al, where RMSE drops from 2.281 to 1.844 wt% and R^2 increases from 0.208 to 0.483 (Figure 5.6).

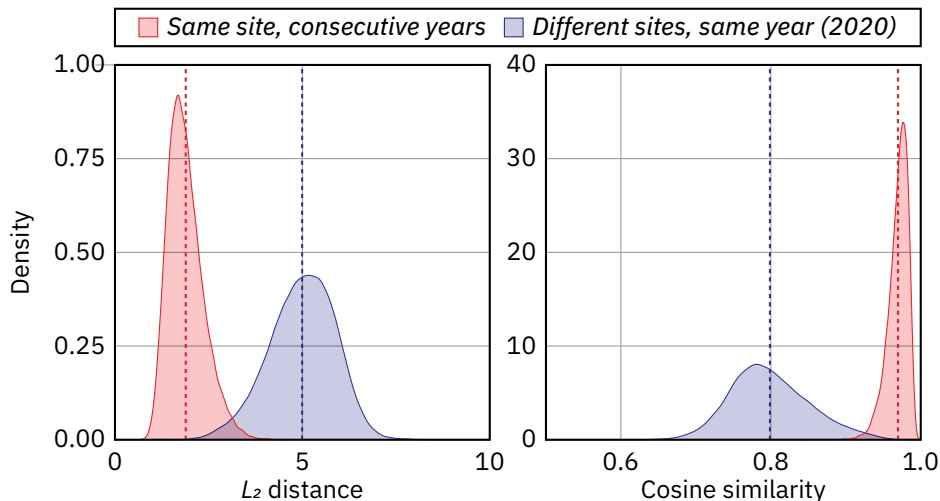


Figure 5.2: Temporal consistency of OlmoEarth Base embeddings via L_2 distance (left) and cosine similarity (right).

Table 5.2: CONUS gains from geology and SSURGO co-variates.

Element	FM	FM + geo/SSURGO	ΔR^2
Al	0.565 ± 0.091	0.624 ± 0.099	+0.059
Ca	0.342 ± 0.097	0.419 ± 0.090	+0.077
Na	0.551 ± 0.074	0.604 ± 0.061	+0.053
K	0.478 ± 0.108	0.506 ± 0.108	+0.028

5.6.2 Discussion

Our experiments reveal several important insights into Geo-FM performance. (1) Frozen OlmoEarth embeddings perform better than raw S2 features when predicting soil element composition, suggesting that generalized models capture useful information for geochemistry tasks even though such efforts are not primary targets of pre-training. (2) The fact that embeddings remain stable over time preempts the potentially thorny question of what embedding set to use for model training. (3 and 4) There is a point at which an increased number of training samples does not improve model performance, but closeness (in embedding space) of a target region to training labels strongly impacts model performance. Taken together, these two points suggest that a global estimate of soil chemistry using Geo-FM may be possible even with the sparse ground truth data that currently exist. (5) Finally, co-variates can help OOD transfer while leaving some errors unresolved, indicating that late fusion may not serve a complete “correction.” The next step may be to include geology

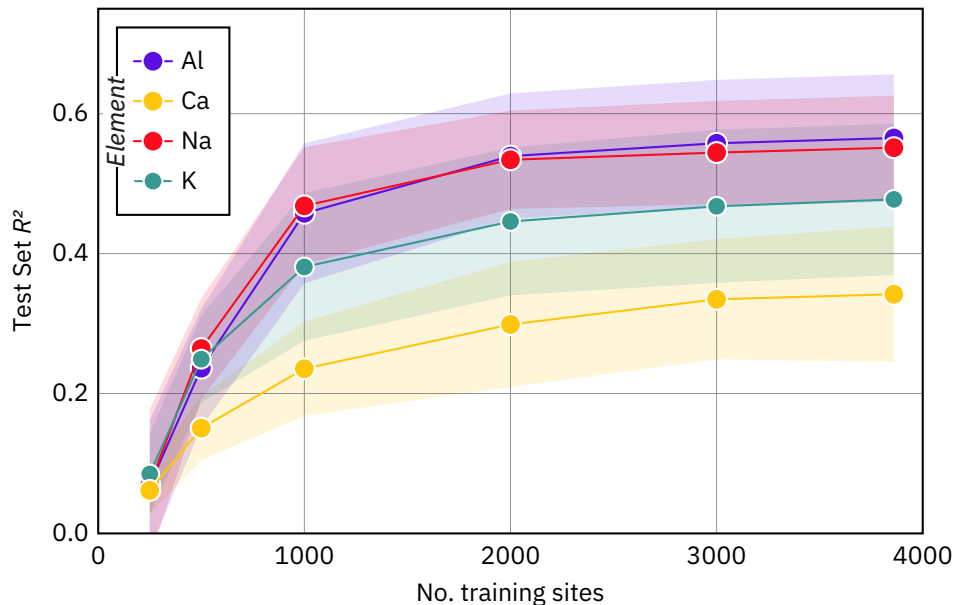


Figure 5.3: Effect of labeled training set size on test-set R^2 . Points show means across 30 runs; shaded bands are $\pm 1SD$.

Table 5.3: Held-out SE coastal performance with and without geology and soil co-variates.

Element	R^2			RMSE ($wt\%$)		
	FM	Geo+soil	Δ	FM	Geo+soil	Δ
Al	0.208	0.483	+0.275	2.281	1.844	-0.437
Ca	0.075	0.213	+0.138	3.194	2.945	-0.249
Na	-0.955	-0.394	+0.561	0.427	0.361	-0.066
K	-0.266	0.015	+0.281	0.844	0.745	-0.099

and soil survey maps earlier in Geo-FM fine-tuning or pretraining, so the model learns from these layers directly instead of receiving them post hoc.

5.7 Conclusion

We evaluate Geo-FM embeddings for soil elemental geochemistry across CONUS using weathering-related elements from the USGS DS-801 survey. OlmoEarth embeddings improve over raw S2 features and remain stable across multiple years, but the results also show that downstream performance depends on training-site density, embedding-space proximity, and regional geology. Adding DS-425 geology and SSURGO co-variates reduces regional holdout error, showing that mapped soil and geology information can help diagnose and improve predictions in regions

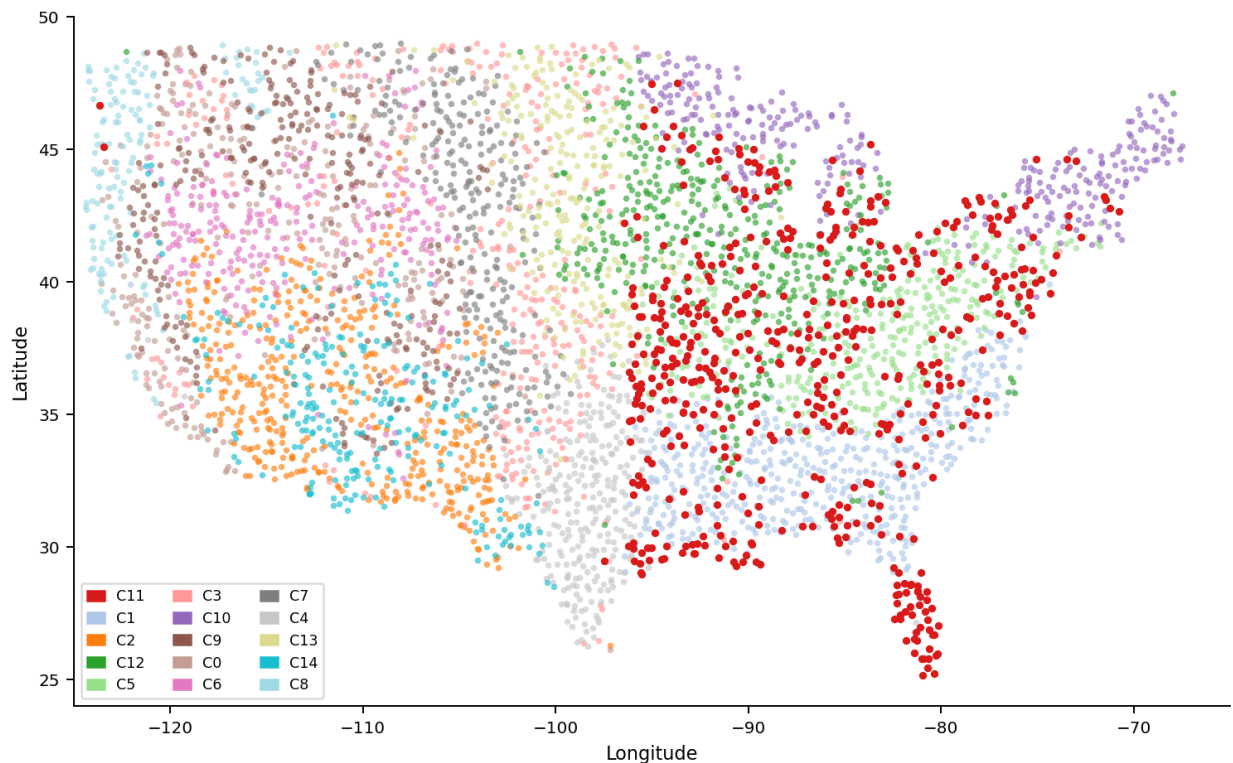


Figure 5.4: The geographic distribution of $k=15$ embedding clusters across 4,814 DS-801 sites. Each color corresponds to one cluster from k -means on L_2 -normalized OlmoEarth Base embeddings.

with few local measurements. More broadly, Geo-FM evaluations for soil geochemistry should test spatial and temporal variation, sample availability, and domain-specific co-variates, so that benchmarks measure reliability under OOD deployment instead of only average accuracy within one region. These lessons are not specific to CONUS, and should guide future Geo-FM soil benchmarks in other regions where field measurements are sparse and geological settings differ from the training data.

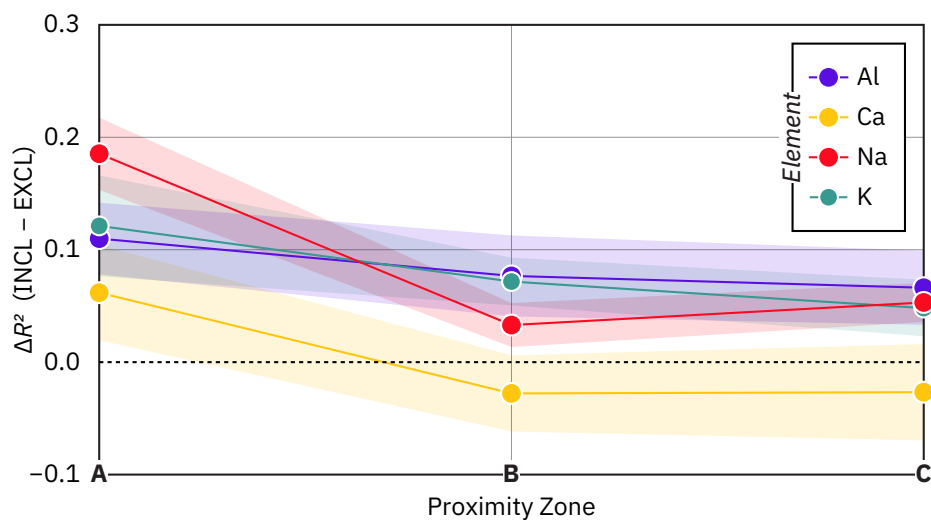


Figure 5.5: Effect of including the focal cluster training sites on nearby non-focal test sites (grouped into proximity zones). Shaded bands show ± 1 SD across seeds.

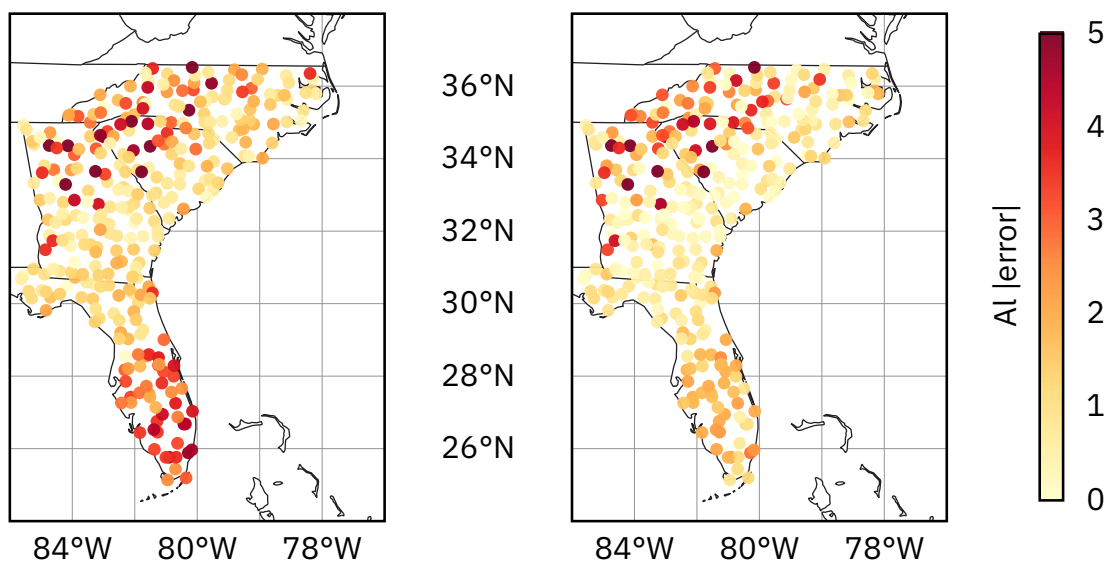


Figure 5.6: Absolute prediction error at SE coastal holdout sites for Al (lower is better). Maps compare the FM-only model (left) with one augmented with co-variates (right).

6 Conclusion

6.1 Summary of the Results

In the Introduction, I identified a reliability gap in the geospatial AI workflow, an incomplete connection between model-centered development and the task-specific information users need. Model outputs are often treated as evidence for a scientific task before that connection is made. This dissertation shows that the gap can be reduced by designing guidance into the workflow, connecting model outputs to the scientific tasks. Each chapter applies this idea at a different stage of the workflow.

Chapter 1 addresses reliability at the image-enhancement stage for object detection. We developed a diffusion model conditioned on numerical speckle filters, trained with a curriculum defined by speckle intensity, and evaluated the model performance by both pixel-level image quality and downstream object detection. The results show that building guidance into the training strategy keeps image enhancement accountable to the downstream task, so despeckling improves image quality without erasing the structures that detection depends on.

Chapter 2 addresses reliability at the validation stage, using lined farm pond mapping in India as the case study. The contribution is an expert-review protocol built directly into the detection workflow. In the Kadwanchi case study, object detections were not treated as final outputs. Instead, each round of detections passed through structured iterative reviews in which experts checked and documented the model’s detections, corrected errors, updated the training data, and recalibrated the resulting pond counts and area estimates. This standardized expert-in-the-loop protocol narrows the reliability gap by converting AI/ML model outputs into metrics and uncertainties that users can apply directly. The same expert corrections also fed back into the training data, improving detection performance itself.

Chapter 3 addresses reliability in soil elemental prediction with Geo-FMs. We used soil chemistry as a test case for applying a general Earth observation embedding to a specialized task

that was not part of its original design. Frozen embeddings were used to predict major soil elements related to chemical weathering, including aluminum, calcium, sodium, and potassium. The analysis showed that soil prediction performance depended on sampling density, proximity in embedding space, and regional geology, while embeddings remained stable across years. Adding geology and soil covariates partly improved performance in the Southeast US coastal holdout, showing that geologic context helps explain variation that image embeddings alone do not capture. The chapter shows that Geo-FM outputs need to be validated across space, time, and geology before they can be treated as reliable for soil prediction.

6.2 Implications

These results show that reliability should be treated as a workflow-level problem. Model training, validation, post-processing, and user interpretation all affect whether an output can support a scientific task. A single strong AI/ML component does not make the full workflow reliable. This view carries three implications for how geospatial AI workflows should be designed.

First, reliability is specific to the task and its users. Technical benchmarks alone cannot establish reliability because users of the same imagery pursue different goals, and each goal requires a different quantity, such as counts, areas, or change over time. What makes a workflow reliable is the connection between its outputs and the people who use them. The connection is most direct when the experts evaluating a model are also the users of its outputs, as in the pond detection case study. Expert involvement, carried out through a transparent and repeatable procedure, gives users a basis for confidence that a benchmark alone cannot provide. When the experts are not the users themselves, the protocol records each review decision so that users can inspect how expert judgment shaped the outputs.

Second, reliability has to be considered at every stage of the workflow, not only at validation. Every component embeds design choices before validation begins, whether tuned in-house against its own measure of quality or inherited from a pretrained tool. A pipeline whose components each perform well by their own measures, such as pixel-level error alone, can still fail the task

as a whole, so component-level metrics, although useful for developing and comparing individual stages, cannot serve as the only design criterion. We should therefore treat alignment with the downstream task as a criterion in its own right, complementing the component metrics.

Third, the need for domain knowledge persists as models become more general. A pretrained model supports many downstream tasks because pretraining optimizes a generic objective, so the workflow carries the responsibility for aligning outputs with a specific scientific task. A common expectation is that generalist models will replace domain expertise. In the applications studied here, domain knowledge remained necessary in workflows. For example, increasing Geo-FM embedding size did not remove the need for domain knowledge outside the model, since geology and soil data improved predictions that model scale alone had left short. Future geospatial AI models, however general, will still require domain knowledge and task-specific validation before users can treat the outputs as reliable evidence.

6.3 Limitations

The main limitation is the applicability of the workflows beyond their demonstrated settings. Each chapter demonstrates its workflow in a single setting chosen for the task. The frameworks themselves are general; the expert-in-the-loop protocol, for example, applies to any geospatial detection task with the same review structure. In contrast, the task-specific components, namely the evaluation metrics and the uncertainty attached to the outputs, follow from each application because reliability depends on the downstream task. I have not yet applied an established framework to a new setting where new users define the metrics, the step that would demonstrate this transfer.

Guidance also carries a cost that likely increases with the scale of the application. Expert review, for example, takes reviewer time and requires consistent judgment across reviewers, which this study did not measure. The broader question is structural, since each guidance mechanism was configured by hand for a single setting, and operational use requires guidance that is organized, documented, and repeatable across sites, dates, and reviewers. Scaling also affects the models, because guidance embedded in a training strategy or a covariate fusion must be reconfigured as data

volumes, regions, and model versions change. The reach of a guided workflow therefore depends on the structure that organizes expertise and domain data as much as on any single component.

The final limitation is the scope of reliability itself. The NIST AI Risk Management Framework treats reliability as one characteristic of trustworthy AI, the base for others that include safety, transparency, and fairness. The workflows in this dissertation have not yet addressed these other characteristics or the consequences of deployment.

6.4 Future Directions

The immediate next steps follow from these limitations. The first is transfer, applying an established framework in a new setting where new users define the metrics. The second is scaling guidance. Active learning can concentrate expert review on the cases where judgment matters most, and larger campaigns will require review protocols evaluated across multiple reviewers. Building on the frameworks established in this dissertation, guidance mechanisms should become reusable components, packaged and documented so that a new application configures them rather than rebuilds them. At the model level, guidance can move into optimization itself, for example by incorporating domain data during pretraining, so that alignment scales with the model rather than being reattached after training.

The broader goal for geospatial AI workflows is trustworthiness. Reliability provides evidence that an output supports the intended task, and trust further depends on whether users understand what an output means in their own context. Meeting this goal requires interpretable outputs, uncertainty that follows the data from the image to the final quantity, and clear statements of the conditions under which a workflow fails. Guided workflows offer a structure for these requirements, because guidance records which stages incorporated domain knowledge and how those choices shaped the output.

Trustworthiness also extends to the models beneath a workflow. Our work relied on OlmoEarth in part because its released weights allow reproducible experiments. Disclosing the data and design choices behind a pretrained model would bring transparency to the model level, giving developers

a basis for judging a representation's suitability for a scientific task before building on it.

This dissertation shows that reliability can be designed into Earth observation AI/ML workflows by connecting models to sensor behavior, expert interpretation, sampling design, and environmental context. Geospatial AI development should treat this connection as a standard element of design, building each workflow around the downstream task and its users. Guided workflows give users justified confidence in the resulting measurements.

References

- Albughdadi, M., Baousis, V., Kaprol, T., Karatosun, A., & Pisa, C. (2024). Exploring SAM transfer learning in optical remote sensing. *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, 2519–2523.
- Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48.
- Bera, S., Gupta, S., Singh, D., & Singh, T. (2025). Automated farm ponds detection using satellite imagery and deep learning: Insights from Kadwanchi village, Maharashtra. *Journal of Geomatics*, 19(1), 108–116.
- Bolya, D., Foley, S., Hays, J., & Hoffman, J. (2020). Tide: A general toolbox for identifying object detection errors. *European Conference on Computer Vision*, 558–573.
- Bommasani, R. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Bostrom, A., Demuth, J. L., Wirz, C. D., Cains, M. G., Schumacher, A., Madlambayan, D., Bansal, A. S., Bearth, A., Chase, R., Crosman, K. M., et al. (2024). Trust and trustworthy artificial intelligence: A research agenda for AI in the environmental sciences. *Risk Analysis*, 44(6), 1498–1513.
- Brown, C. F., Kazmierski, M. R., Pasquarella, V. J., Rucklidge, W. J., Samsikova, M., Zhang, C., Shelhamer, E., Lahera, E., Wiles, O., Ilyushchenko, S., et al. (2025). AlphaEarth foundations: An embedding field model for accurate and efficient global mapping from sparse label data. *arXiv preprint arXiv:2507.22291*.
- Chierchia, G., Cozzolino, D., Poggi, G., & Verdoliva, L. (2017). Sar image despeckling through convolutional neural networks. *2017 IEEE international geoscience and remote sensing symposium (IGARSS)*, 5438–5441.

- Dalsasso, E., Denis, L., & Tupin, F. (2022). As if by magic: Self-supervised training of deep despeckling networks with merlin. *IEEE Transactions on Geoscience and Remote Sensing*, 60(1), 1–13.
- Devillers, R., Stein, A., Bédard, Y., Chrisman, N., Fisher, P., & Shi, W. (2010). Thirty years of research on spatial data quality: Achievements, failures, and opportunities. *Transactions in GIS*, 14(4), 387–400.
- Dominati, E., Patterson, M., & Mackay, A. (2010). A framework for classifying and quantifying the natural capital and ecosystem services of soils. *Ecological Economics*, 69(9), 1858–1868.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*, 88(2), 303–338.
- Ghorbanzadeh, O., Xu, Y., Ghamisi, P., Kopp, M., & Kreil, D. (2022). Landslide4sense: Reference benchmark data and deep learning models for landslide detection. *arXiv preprint arXiv:2206.00515*.
- Gil, Y., Honaker, J., Gupta, S., Ma, Y., D’Orazio, V., Garijo, D., Gadewar, S., Yang, Q., & Jahanshad, N. (2019). Towards human-guided machine learning. *Proceedings of the 24th international conference on intelligent user interfaces*, 614–624.
- Goodchild, M. F., & Li, W. (2021). Replication across space and time must be weak in the social and environmental sciences. *Proceedings of the National Academy of Sciences*, 118(35), e2015759118.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.
- Goodman, J. W. (1976). Some fundamental properties of speckle. *Journal of the Optical Society of America*, 66, 1145–1150.
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

- Hengl, T., Miller, M. A., Križan, J., Shepherd, K. D., Sila, A., Kilibarda, M., Antonijević, O., Glušica, L., Dobermann, A., Haefele, S. M., et al. (2021). African soil properties and nutrients mapped at 30 m spatial resolution using two-scale ensemble machine learning. *Scientific Reports*, 11(1), 6130.
- Herrera-Poyatos, A., Del Ser, J., de Prado, M. L., Wang, F.-Y., Herrera-Viedma, E., & Herrera, F. (2025). Responsible artificial intelligence systems: A roadmap to society's trust through trustworthy AI, auditability, accountability, and governance. *arXiv preprint arXiv:2503.04739*.
- Herzog, H., Bastani, F., Zhang, Y., Tseng, G., Redmon, J., Sablon, H., Park, R., Morrison, J., Buraczynski, A., Farley, K., et al. (2025). OlmoEarth: Stable latent image modeling for multimodal earth observation. *arXiv preprint arXiv:2511.13655*.
- Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33, 6840–6851.
- Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., & Salimans, T. (2022). Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47), 1–33.
- Huang, B., Zhao, B., & Song, Y. (2018). Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment*, 214, 73–86.
- Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., & Yang, H. (2019). Machine learning and artificial intelligence to aid climate change research and preparedness. *Environmental Research Letters*, 14(12), 124007.
- Jenny, H. (1994). *Factors of soil formation: A system of quantitative pedology*. Courier Corporation.
- Kale, E. (2017). Problematic uses and practices of farm ponds in Maharashtra. *Economic and Political Weekly*, 20–22.
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3), 312–322.

- Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., & Kumar, V. (2017). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10), 2318–2331.
- Kaur, D., Uslu, S., Rittichier, K. J., & Durrezi, A. (2022). Trustworthy artificial intelligence: A review. *ACM computing surveys (CSUR)*, 55(2), 1–38.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.
- Ko, J., & Lee, S. (2021). Sar image despeckling using continuous attention module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 3–19.
- Kuan, D. T., Sawchuk, A. A., Strand, T. C., & Chavel, P. (1985). Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2), 165–177.
- Lee, J.-S. (1981). Speckle analysis and smoothing of synthetic aperture radar images. *Computer Graphics and Image Processing*, 17(1), 24–32.
- Li, Y., Li, X., Li, W., Hou, Q., Liu, L., Cheng, M.-M., & Yang, J. (2024). Sardet-100k: Towards open-source benchmark and toolkit for large-scale sar object detection. *Advances in Neural Information Processing Systems*, 37, 128430–128461.
- Lu, S., Guo, J., Zimmer-Dauphinee, J. R., Nieusma, J. M., Wang, X., Wernke, S. A., Huo, Y., et al. (2025). Vision foundation models in remote sensing: A survey. *IEEE Geoscience and Remote Sensing Magazine*.
- Maxar Technologies. (2020a). WorldView-2 datasheet [MXR-DS-wv2 06/20]. <https://read.uberflip.com/i/1365577>
- Maxar Technologies. (2020b). WorldView-3 datasheet [MXR-DS-wv3 08/20]. <https://read.uberflip.com/i/1365580>

- McGovern, A., Bostrom, A., Davis, P., Demuth, J. L., Ebert-Uphoff, I., He, R., Hickey, J., Gagne II, D. J., Snook, N., Stewart, J. Q., et al. (2022). NSF AI institute for research on trustworthy AI in weather, climate, and coastal oceanography (AI2ES). *Bulletin of the American Meteorological Society*, 103(7), E1658–E1668.
- McGovern, A., Demuth, J., Bostrom, A., Wirz, C. D., Tissot, P. E., Cains, M. G., & Musgrave, K. D. (2024). The value of convergence research for developing trustworthy AI for weather, climate, and ocean hazards. *npj Natural Hazards*, 1(1), 13.
- McGovern, A., Ebert-Uphoff, I., Gagne II, D. J., & Bostrom, A. (2022). Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environmental Data Science*, 1, e6.
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7), 3523–3542.
- Minasny, B., & McBratney, A. B. (2016). Digital soil mapping: A brief history and some lessons. *Geoderma*, 264, 301–311.
- Mondini, A. C., Guzzetti, F., Chang, K.-T., Monserrat, O., Martha, T. R., & Manconi, A. (2021). Landslide failures detection and mapping using synthetic aperture radar: Past, present and future. *Earth-Science Reviews*, 216, 103574.
- Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., & Fernández-Leal, Á. (2023). Human-in-the-loop machine learning: A state of the art. *Artificial Intelligence Review*, 56(4), 3005–3054.
- Mulder, V., De Bruin, S., Schaepman, M. E., & Mayr, T. (2011). The use of remote sensing in soil and terrain mapping—a review. *Geoderma*, 162(1-2), 1–19.
- National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (tech. rep. No. NIST AI 100-1). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.AI.100-1>

- Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *International conference on machine learning*, 8162–8171.
- Olofsson, P., Foody, G. M., Herold, M., Stehman, S. V., Woodcock, C. E., & Wulder, M. A. (2014). Good practices for estimating area and assessing accuracy of land change. *Remote Sensing of Environment*, 148, 42–57.
- Osco, L. P., Wu, Q., De Lemos, E. L., Gonçalves, W. N., Ramos, A. P. M., Li, J., & Junior, J. M. (2023). The segment anything model (SAM) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation*, 124, 103540.
- Parrilli, S., Poderico, M., Angelino, C. V., & Verdoliva, L. (2011). A nonlocal sar image denoising algorithm based on lmmse wavelet shrinkage. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2), 606–616.
- Paul, A., & Savakis, A. (2025). On denoising diffusion probabilistic models for synthetic aperture radar despeckling. *Sensors*, 25(7), 2149.
- Pawar, C., Bolage, R., & Ghalme, S. (2024). Assessment of crop diversification index (CDI)-in Kadwanchi village of Jalana District-a success story of sustainable development principles. *Journal of Advanced Zoology*, 45.
- Perera, M. V., Nair, N. G., Bandara, W. G. C., & Patel, V. M. (2023). Sar despeckling using a denoising diffusion probabilistic model. *IEEE Geoscience and Remote Sensing Letters*, 20, 1–5.
- Prasad, P., & Sohoni, M. A. (2018). Farmponds for horticulture: Boon or curse? analysing impact on farm profitability, resource sustainability and social welfare. *Proc., 36th Intl. Conf. of the System Dynamics Society*.
- Rolf, E. (2023). Evaluation challenges for geospatial ml. *arXiv preprint arXiv:2303.18087*.
- Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Slavin Ross, A., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A. S., Maharaj, T., Sherwin, E. D., Mukkavilli, S. K., Kording, K. P., Gomes, C. P., Ng, A. Y., Hassabis, D.,

- Platt, J. C., . . . Bengio, Y. (2022). Tackling climate change with machine learning. *ACM Computing Surveys*, 55(2), 1–96. <https://doi.org/10.1145/3485128>
- Shah, S. H., Harris, L. M., Johnson, M. S., & Wittman, H. (2021). A ‘drought-free’ Maharashtra? politicising water conservation for rain-dependent agriculture. *Water Alternatives*, 14(2), 573–596.
- Sidhu, B. S., & Shah, S. H. (2021). *Quantifying evaporation loss from farm ponds: A case from Jalna District, Maharashtra* [Preprint]. <https://doi.org/10.13140/RG.2.2.12724.99200>
- Smith, D. B., Cannon, W. F., Woodruff, L. G., Solano, F., Kilburn, J. E., & Fey, D. L. (2013). *Geochemical and mineralogical data for soils of the conterminous united states* (tech. rep.). US Geological Survey.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning*, 2256–2265.
- Soller, D. R., Reheis, M. C., Garrity, C. P., & Van Sistine, D. R. (2009). Map database for surficial materials in the conterminous united states. *US Geological Survey Data Series*, 425.
- Song, J., Zhao, C., Liao, H.-Y., & Shao, W. (2025). Foundationsoil: Enhancing soil organic carbon mapping using a multi-temporal geospatial foundation model. *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, 1067–1070.
- Stehman, S. V., & Foody, G. M. (2019). Key issues in rigorous accuracy assessment of land cover products. *Remote Sensing of Environment*, 231, 111199.
- Szwarcman, D., Roy, S., Fraccaro, P., Gíslason, E., Blumenstiel, B., Ghosal, R., De Oliveira, P. H., de Sousa Almeida, J. L., Sedona, R., Kang, Y., et al. (2025). Prithvi-EO-2.0: A versatile multi-temporal foundation model for earth observation applications. *IEEE Transactions on Geoscience and Remote Sensing*.
- Tay, C. W. J., Yun, S.-H., Chin, S. T., Bhardwaj, A., Jung, J., & Hill, E. M. (2020). Rapid flood and damage mapping using synthetic aperture radar in response to typhoon hagibis, japan. *Scientific Data*, 7(1), 100.

- Tundia, C., Kumar, R., Damani, O., & Sivakumar, G. (2023). FPCD: An open aerial VHR dataset for farm pond change detection. *arXiv preprint arXiv:2302.14554*.
- Tyukavina, A., Stehman, S. V., Foody, G. M., Bontemps, S., See, L., Olofsson, P., Tsensbazar, N., Radoux, J., Komarova, A., Serre, B. M., Song, X., d'Andrimont, R., Koren, G., Potapov, P., Bullock, E. L., Campbell, P., de Bruin, S., Defourny, P., Friedl, M. A., . . . Xiao, X. (2025). Land cover and change map accuracy assessment and area estimation good practices protocol. version 1.0 - 2025.
- Varghese, R., & Sambath, M. (2024). YOLOv8: A novel object detection algorithm with enhanced performance and robustness. *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, 1–6.
- Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al. (2021). Informed machine learning—a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1), 614–633.
- Wagstaff, K. (2012). Machine learning that matters. *arXiv preprint arXiv:1206.4656*.
- Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4), 1–37.
- Xia, J., Yokoya, N., Adriano, B., & Broni-Bediako, C. (2023). Openearthmap: A benchmark dataset for global high-resolution land cover mapping. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6254–6264.
- Zhang, C., Yue, P., Tapete, D., Jiang, L., Shangguan, B., Huang, L., & Liu, G. (2020). A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166, 183–200.
- Zhang, T., Zhang, X., Li, J., Xu, X., Wang, B., Zhan, X., Xu, Y., Ke, X., Zeng, T., Su, H., Ahmad, I., Pan, D., Liu, C., Zhou, Y., Shi, J., & Wei, S. (2021). Sar ship detection dataset (ssdd): Official release and comprehensive data analysis. *Remote Sensing*, 13(18), 3690.

- Zhang, Y., Mehta, S., & Caspi, A. (2021). Rethinking semantic segmentation evaluation for explainability and model selection. *arXiv preprint arXiv:2101.08418*.
- Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine*, 5(4), 8–36.
- Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3), 257–276.