SPECTRA: Band-Routed Embedding and Stage-Wise LoRA for Cross-Sensor Fine-Tuning of Geospatial Foundation Models

Xingyan Li
xingyanli@umbc.edu
University of Maryland, Baltimore County
Baltimore, Maryland, USA

Jordan A. Caraballo-Vega
jordan.a.caraballo-vega@nasa.gov
NASA Goddard Space Flight Center
Greenbelt, Maryland, USA

Jie Gong
jie.gong@nasa.gov
NASA Goddard Space Flight Center
Greenbelt, Maryland, USA

Mark L. Carroll
mark.carroll@nasa.gov
NASA Goddard Space Flight Center
Greenbelt, Maryland, USA

Jianwu Wang
jianwu@umbc.edu
Univ. of Maryland, Baltimore County
Baltimore, Maryland, USA

Abstract

Geospatial foundation models (GeoFMs), pretrained on large-scale geospatial data such as Earth observation (EO), climate, and weather data, have shown promising performance when fine-tuned on diverse downstream tasks. However, there are two challenges of adapting EO-pretrained GeoFMs to practical downstream datasets. The first challenge is how to handle spectral mismatch: pretrained patch embeddings expect a fixed set of input bands, whereas downstream sensors may provide different channels. The second challenge is how to reduce fine-tuning cost and make it efficient. While existing work has made efforts on these challenges individually, jointly improving fine-tuning performance under spectral mismatch while reducing adaptation cost remains underexplored. We propose SPECTRA, a parameter-efficient fine-tuning framework that addresses both spectral mismatch and adaptation cost. To handle spectral mismatch, SPECTRA introduces Band-Routed Embedding (BRE), which maps all available downstream bands into the band space expected by the pretrained GeoFM. By using BRE, all available bands in the downstream dataset are utilized to improve the selected-band input without changing the pretrained patch embedding interface. To reduce adaptation cost, SPECTRA further introduces a Stage-wise Transferability-aware LoRA (ST-LoRA) fine-tuning. ST-LoRA estimates stage-wise transferability before fine-tuning and assigns stage-specific LoRA ranks, concentrating trainable parameters on the stages with high transferability for the target task. Across three EO-pretrained GeoFMs and four downstream segmentation datasets, experiments show that BRE improves performance by utilizing all spectral bands, while ST-LoRA reduces trainable parameters compared with full fine-tuning and standard LoRA. Code is available at `github.com/big-data-lab-umbc/SPECTRA`.

CCS Concepts

• **Computing methodologies** → **Computer vision tasks**; **Transfer learning**; • **Applied computing** → *Earth and atmospheric sciences.*

Keywords

geospatial foundation models, cross-sensor adaptation, parameter-efficient fine-tuning, LoRA, multispectral imagery

ACM Reference Format:
Xingyan Li, Jordan A. Caraballo-Vega, Jie Gong, Mark L. Carroll, and Jianwu Wang. 2026. SPECTRA: Band-Routed Embedding and Stage-Wise LoRA for Cross-Sensor Fine-Tuning of Geospatial Foundation Models. In *Proceedings of the 34th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '26).* ACM, New York, NY, USA, 13 pages.

1 Introduction

Geospatial foundation models (GeoFMs) are increasingly used as reusable backbones for Earth-observation (EO) tasks. Models such as Prithvi, Prithvi-EO-2.0, SatMAE, and ScaleMAE pretrain large backbones on satellite imagery and then fine-tune the learned encoder to downstream dense EO prediction problems [3, 13, 24, 26]. Many of these models follow a masked-autoencoding-style pretraining workflow: input images are divided into patches, projected into tokens by a *patch-embedding layer*, processed by a *vision transformer (ViT) encoder*, and reconstructed by a decoder used only during pretraining [6, 10]. After pretraining, the pretrained patch embedding and the encoder are reused for downstream fine-tuning, and a new decoder is added for downstream tasks such as segmentation. This pretrain-then-fine-tune workflow is efficient for EO because dense geospatial labels are expensive to obtain, and downstream datasets are often limited in geography, time, sensor type, or event coverage [15, 19, 20]. However, there are two challenges of adapting EO-pretrained GeoFMs to practical downstream datasets.

A first challenge is spectral mismatch. The pretrained patch embedding defines a fixed input band interface: it expects a particular number, order, and spectral meaning of input channels. However, downstream EO datasets may come from different sensors, contain additional multispectral bands. A common compatibility strategy, denoted as band-selection in this paper, is to select the downstream bands whose physical characteristics (such as wavelengths) best match the pretrained input bands, while discarding channels that cannot be matched. Similar band-matching strategies are used in recent GeoFM fine-tuning works to make heterogeneous downstream datasets compatible with fixed-interface pretrained models [20]. This strategy preserves the pretrained patch embedding and provides a stable, physically-motivated baseline. However, even when band-selection selects a target band for a pretrained input slot, discarded channels may contain task-relevant information. An alternative is to redesign or adapt the input embedding so that heterogeneous or multisensor inputs can be consumed directly, as in wavelength-conditioned, multisensor, or any-sensor EO foundation models [1, 9, 29]. These approaches increase spectral flexibility, but they may also introduce additional trainable parameters during adaptation and thus increase adaptation cost.

A second challenge is fine-tuning cost. Different fine-tuning strategies can be different in computing cost. Full fine-tuning updates all parameters of a large pretrained encoder, which can be expensive in GPU memory, optimization time, and model storage. Parameter-efficient fine-tuning (PEFT) methods reduce this cost by freezing most pretrained weights and training only a small number of task-specific parameters [11]. LoRA, which injects low-rank trainable matrices into existing linear layers, [12], is one of the most attractive PEFT methods because it is easy to implement and effective. Recent work on EO and GeoFM adaptation also shows that PEFT methods can provide competitive performance while reducing trainable parameters, training time, and memory requirements [4, 21]. However, standard LoRA can be further improved because it usually applies the same rank across layers or stages, even though different encoder stages may have different transferability to a target task. Another PEFT method, Surgical fine-tuning, shows that selectively adapting a subset of pretrained layers can improve transfer under distribution shift [16], but unfreezing an entire stage can still require many more trainable parameters than LoRA. We compare the full fine-tuning, LoRA, and Surgical fine-tuning and our proposed idea in Figure 1.

For both of the two challenges, although prior work has made remarkable progress on spectral adaptation and parameter-efficient fine-tuning separately, jointly addressing both challenges, to the best of our knowledge, still needs to be explored. A practical cross-sensor GeoFM PEFT method should use the spectral information available in downstream inputs. At the same time, it should adapt the encoder under a controlled trainable-parameter budget.

To handle the two problems, we propose **SPECTRA**, a parameter-efficient cross-sensor fine-tuning framework that solves the spectral mismatch challenge by input spectral adaptation and solves the cost challenge by stage-wise encoder adaptation. The framework contains two key components: Band-Routed Embedding (BRE) and Stage-wise Transferability-aware LoRA (ST-LoRA). BRE maps all available downstream bands into the band space expected by the pretrained model, producing a "virtual image" compatible to pre-training input. It preserves the selected-band input physically grounded anchor and learns routing-gated residual contributions from all fine-tuning bands, allowing information beyond the selected subset to refine the virtual pretrained-compatible image. Stage-wise Transferability-aware LoRA (ST-LoRA) addresses encoder adaptation cost by partitioning the pretrained encoder into stages and assigning stage-specific LoRA ranks before fine-tuning. It contains a transferability-aware planner (**ST planner**) that estimates stage-wise transferability using frozen-feature metrics such as LogME [30], then allocates LoRA capacity based on the LogME profile. The resulting pipeline aims to learn which stage is useful to learn spectral information and assign trainable LoRA ranks accordingly, instead of using standard uniform LoRA ranks.

Our experiments evaluate SPECTRA across three EO-pretrained GeoFMs and four downstream segmentation datasets, covering

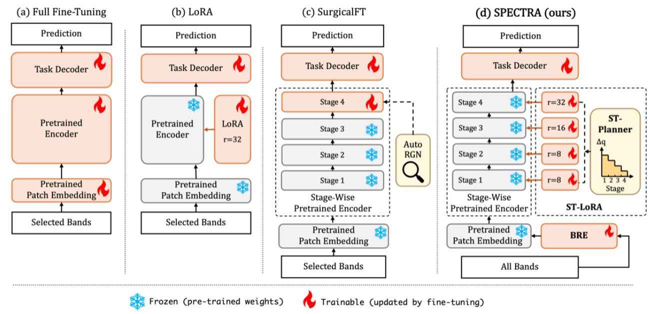

Figure 1: Comparison of PEFT methods compared in this work. SPECTRA adds a pretrained-compatible BRE input path and uses ST-LoRA to assign stage-wise LoRA ranks.

both spectrally aligned and mismatched transfer settings. Across five fine-tuning policies and 60 matched backbone–dataset–policy configurations, adding BRE improves mean macro mIoU by +2.88 points over the corresponding band-selection baseline, whereas a direct MLP projector underperforms band-selection. The gains are most pronounced when band selection discards substantial downstream spectral information and are smaller when the pretrained and downstream band interfaces are already well aligned. In stage-wise adaptation on Sen1Floods11, SPECTRA also outperforms the LoRA-32 band-selection baseline across all three backbones while reducing active trainable parameters by 17.1–33.6%. Comparisons with manually selected rank schedules further demonstrate that ST-LoRA provides effective automatic rank allocation under a limited parameter budget.

This paper makes three contributions with the proposed SPECTRA method:

- We introduce **Band-Routed Embedding (BRE)**, a lightweight input adaptation module that learns source-to-target band contributions for spectral mismatch between pre-training and fine-tuning datasets. BRE learns routing-gated contributions from all fine-tuning bands, allowing information beyond the selected subset to refine the selected-band input without modifying the pretrained patch embedding.
- We introduce **Stage-wise Transferability-aware LoRA (ST-LoRA)** for parameter-efficient GeoFM adaptation. ST-LoRA has an ST-planner that estimates the transferability of each pretrained encoder stage before fine-tuning and assigns stage-specific LoRA ranks. By concentrating trainable parameters in stages with high transferability and reducing parameters in other stages, ST-LoRA helps to reduce the number of trainable parameters compared to standard parameter-efficient LoRA.
- We conduct a comprehensive evaluation across three EO-pretrained GeoFMs and four downstream segmentation datasets with different spectral configurations. The experiments include quantitative accuracy comparisons, trainable-parameter accounting, BRE ablation study, rank-schedule analyses, and

qualitative visualization examples. Not only do the quantitative results show our methods are effective, but the visualization examples also illustrate that SPECTRA helps capture details in the images.

2 Related Work

Geospatial foundation models. Remote-sensing foundation models adapt masked autoencoding and transformer pretraining to imagery with spectral, spatial, and temporal structure. SatMAE introduces temporal and spectral positional structure for satellite imagery [3]; ScaleMAE injects scale awareness for multiscale geospatial representation learning [24]; and Prithvi-style models pretrain large geospatial transformers on HLS-like imagery [13, 26]. These models differ not only in architecture and pretraining data, but also in their assumed input sensors and native bands. SPECTRA targets this interface mismatch rather than treating the backbone as a generic image encoder.

Parameter-efficient fine-tuning. Fine-tuning all backbone parameters is a strong reference point, but it is expensive for large GeoFMs and can be difficult to justify when downstream labels are limited. LoRA injects low-rank updates into frozen layers and controls trainable capacity through the rank [12]. Uniform LoRA ranks are simple, but they assume every encoder stage needs the same adaptation capacity. SPECTRA keeps LoRA as the encoder adaptation mechanism and studies whether rank should instead vary by stage. The differences of these approaches are illustrated in Figure 1.

Cross-sensor adaptation. Sensor mismatch is common in EO workflows: a downstream task may use a different satellite, a different subset of optical bands, SAR-derived channels, elevation features, or task-specific stacks. Band selection is robust because it keeps the pretrained interface unchanged, but it ignores measurements outside the matched band set. Learned projections and embedding replacement can ingest arbitrary bands, but they also move the input distribution seen by the pretrained encoder. BRE is designed around a conservative principle: keep the pretrained patch embedding as the tokenizer and learn only a small virtual-band correction in image space.

Transferability diagnostics and rank allocation. Transferability metrics such as LEEP [23] and LogME [30] estimate how useful pretrained features are before full fine-tuning. In our setting, these diagnostics serve two roles. First, they help explain why band-selection performance differs across backbone-dataset pairs. Second, they provide a low-cost signal for allocating LoRA rank across encoder stages. ST-LoRA uses this diagnostic view for GeoFM fine-tuning, where both spectral mismatch and task transferability matter.

3 Background

3.1 Problem Statement

Let $X \in \mathbb{R}^{C\times H\times W}$ denote a target image with C available channels and dense labels Y. A pretrained GeoFM has a pre-trained patch embedding E_0 that expects K compatible input bands. In cross-sensor fine-tuning, usually $C \neq K$ or the C target channels do not match the spectral interface used during pretraining. The band-selection baseline resolves this mismatch by selecting a subset $\mathcal{S} \subseteq \{1,\ldots,C\}$ with $|\mathcal{S}| = K$:

$$X_{\text{sel}} = P_{\text{sel}}(X) \in \mathbb{R}^{K\times H\times W}, \tag{1}$$

where P_{sel} selects the K target channels whose central wavelengths or metadata best match the pretrained interface. This keeps the original patch embedding valid, but it makes every unselected channel invisible.

The first problem is therefore input adaptation: construct a pretrained-compatible map that can use all observed target channels while still producing a K-channel input for E_0. We write this desired map as

$$A_\theta : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{K\times H\times W}, \qquad X_{\text{all}} = A_\theta(X). \tag{2}$$

In this paper, "fully using the bands" means that no target channel is hard-discarded by the input map. Formally, every channel can influence at least one virtual pretrained band. This condition does not claim that every band is always useful for every task; it states the structural requirement that all bands remain available to the fine-tuned model.

The second problem is adaptation cost. Let $F_{\omega,\phi}$ denote a pretrained encoder with frozen backbone weights ω and trainable adaptation (e.g. LoRA weights) parameters ϕ, and let D_η denote the segmentation decoder with trainable parameters η. A fine-tuned prediction has the form $\hat{Y} = D\big(F_{\omega,\phi}(E_0(A_\theta(X)))\big)$. The training objective is standard supervised segmentation to minimize the loss $\mathcal{L}$ between prediction $\hat{Y}$ and true labels Y: $\min_{\theta,\phi,\eta} \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}\Big(\hat{Y}_n, Y_n\Big)$. Our design goal is to keep the trainable percentage ρ small.

$$\rho = \frac{|\theta| + |\phi| + |\eta|}{|\omega| + |\theta| + |\phi| + |\eta|}, \tag{3}$$

To tackle the two problems, we **focus on improving A_θ and ϕ.** **For the first input adaptation problem**, we propose a new input adapter A_θ. **For the second problem**, we propose a parameter-efficient fine-tuning method to decrease ρ by reducing the number ϕ. In this paper, band-selection is the default input adapter A_θ for the baseline methods; BRE is our proposed all-band input adaptation module.

3.2 Models and Datasets

We use three GeoFMs and four downstream datasets to evaluate spectral transfer under different levels of input-band mismatch. Table 1 summarizes the pretrained model. For each GeoFM, it reports the pretraining data source, the fixed input band interface, the number of bands K expected by the pretrained patch embedding, and the ViT backbone size. Table 2 summarizes the downstream datasets. For each target dataset, it reports the input data source, the full number of available target channels C, the number of semantic classes, and the prediction task. Together, these tables define the 12 backbone–dataset combinations used in our experiments and show where spectral mismatch can arise. For example, Prithvi-EO-2.0 has a six-band HLS optical interface, ScaleMAE has an RGB-only interface, and SatMAE has a ten-band Sentinel-2 multispectral interface, while the downstream datasets provide between 6 and 14 input channels. Full spectral-band metadata for the pretrained interfaces and downstream datasets are provided in Appendix Tables 7 and 8.

For all GeoFMs and datasets, the definition band-selection method for our study is important because the it is used in both baseline

Table 1: GeoFM backbones, pretrained data sources, and spectral interfaces.

GeoFM	Pretrained data source	Input interface	Bands (K)	Backbone
Prithvi-EO-2.0	HLS (Landsat/Sentinel-2) [26]	HLS optical	6	ViT-Huge
SatMAE	fMoW-Sentinel / Sentinel-2 [3]	Sentinel-2 multispectral	10	ViT-Large
ScaleMAE	fMoW/ImageNet RGB [24]	RGB	3	ViT-Large

Table 2: Downstream datasets, target input sources, and full target-channel counts.

Dataset	Target input source	Bands (C)	Classes	Task
Sen1Floods11	Sentinel-2 / Sen1Floods11 [2]	13	2	Flood
FireScars	HLS FireScars / GEO-Bench [15]	6	2	Burn-scar
Landslide4Sense	Sentinel-2 + DEM channels [7]	14	2	Landslide
SA Crop Type	Sentinel-2 L2A / GEO-Bench [15]	12	10	Crop-type

methods and our proposed adaptation module. We define the band-selection method as Equation 1. Also, Table 3 summarizes how downstream target bands are mapped to the K pretrained input slots for each GeoFM–dataset pair. The selected-index column lists the target channel indices fed to the pretrained patch embedding, in pretrained input-slot order. Repeated indices indicate that one target band is reused for multiple pretrained slots. Target bands that are not selected for any pretrained slot are marked as discarded for that GeoFM–dataset pair. Auxiliary channels, such as DEM, are not wavelength-matched and are treated as non-spectral auxiliary inputs. We also report a LogME-based [30] transfer gap of frozen pretrained encoder. Lower values indicate stronger transferability, suggesting that the frozen representation is already well aligned with the downstream task. Full band-name mappings are provided in Appendix Tables 9, 10, and 11.

Together, these settings test three different spectral-transfer regimes using three GeoFMs. With **Prithvi-EO-2.0**, the downstream inputs are close to the HLS optical interface used during pretraining. In this regime, band-selection can select K highly matched channels from the target input X, and BRE mainly tests whether the remaining fine-tuning bands provide useful residual information beyond this strong selected-band baseline. Prithvi–FireScars is the clearest native-match case, since both the pretrained interface and the downstream dataset use six HLS-like optical bands. With **ScaleMAE**, the pretrained interface is much narrower because the pre-trained model expects only RGB inputs, and the RGB wavelengths are not always exactly aligned with the target multispectral bands. This creates an information bottleneck for band-selection because it must discard most non-RGB channels; BRE is therefore expected to allows all fine-tuning bands to contribute to the input. With **SatMAE**, the pretrained interface is richer than RGB but still not perfectly matched to every downstream sensor interface. Although band-selection maps the target channels to the pretrained K slots using the closest available central wavelengths, some slots may only match approximately and even have repeated target bands, leaving a larger spectral gap for BRE to compensate. These three cases therefore evaluate BRE under native HLS-compatible, RGB-limited, and multispectral-but-misaligned transfer settings.

Table 3: Band-selection and transferability diagnostics for all GeoFM–dataset pairs in this study.

Backbone	Dataset	Selected band indexes	Discarded band indexes	Transfer gap ↓
Prithvi	Sen1Floods11	[1, 2, 3, 8, 11, 12]	[0, 4, 5, 6, 7, 9, 10]	0.223
	FireScars	[0, 1, 2, 3, 4, 5]	–	0.401
	Landslide4Sense	[1, 2, 3, 8, 11, 12]	[0, 4, 5, 6, 7, 9, 10, 13]	0.338
	SA Crop Type	[1, 2, 3, 8, 10, 11]	[0, 4, 5, 6, 7, 9]	0.545
ScaleMAE	Sen1Floods11	[3, 2, 1]	[0, 4, 5, 6, 7, 8, 9, 10, 11, 12]	0.555
	FireScars	[2, 1, 0]	[3, 4, 5]	0.748
	Landslide4Sense	[3, 2, 1]	[0, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]	0.000
	SA Crop Type	[3, 2, 1]	[0, 4, 5, 6, 7, 8, 9, 10, 11]	0.541
SatMAE	Sen1Floods11	[1, 2, 3, 4, 5, 6, 7, 8, 11, 12]	[0, 9, 10]	0.194
	FireScars	[0, 1, 2, 2, 2, 3, 3, 3, 4, 5]	–	0.531
	Landslide4Sense	[1, 2, 3, 4, 5, 6, 7, 8, 11, 12]	[0, 9, 10, 13]	0.203
	SA Crop Type	[1, 2, 3, 4, 5, 6, 7, 8, 10, 11]	[0, 9]	0.580

4 Method

SPECTRA has two proposed components: BRE and ST-LoRA, and Figure 2 illustrates how the two components work with other modules in the fine-tuning pipeline. To begin with, BRE converts the full target image $X \in \mathbb{R}^{C\times H\times W}$ into a virtual input image $X_{BRE} \in \mathbb{R}^{K\times H\times W}$ that is compatible with K-band pretrained input images. It starts from the band-selection anchor, learns band-routed gates over all observed bands, and applies a residual adapter to learn a projection so that the resulting X_{BRE} is the same shape as the input image expected by the pretrained patch embedding E_0 of the GeoFM. The virtual input image is learned by E_0 xxx embeded tokens for pretrained encoder of the GeoFM. The pretrained encoder is divided into several (e.g. four) stages, and each encoder stage is adapted by LoRA which is also divided into the same stages, where each stage has its own LoRA rank. For the stage-wise LoRA, ST-LoRA estimates stage-wise transferability to estimate stage-wise LoRA ranks for the encoder. After the encoder learns xx, the prediction is finally produced by the UPerNet decoder.

4.1 Band-Routed Embedding

BRE is our input adaptation module for spectral mismatch. As defined in the problem statement, the goal of input adaptation is to construct a map $A_\theta : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{K\times H\times W}$ that uses the full target image X while still producing a K-channel image compatible with the pretrained patch embedding E_0. In our method, BRE implements this adapter. We denote its output as

$$X_{\mathrm{BRE}} = A_\theta(X) \in \mathbb{R}^{K\times H\times W}.$$

As illustrated in Figure 2, there are two paths derived from the input images: a selected-band path (the upper path in the figure), and a residual path (the lower path in the figure) adapting all bands as input with band routed gates and a residual adapter. The key design principle is to keep the selected-band input as a physically grounded anchor and to add a learned residual correction from all available target bands.

Selected-band anchor. Let $\mathcal{S} = (s_1, \ldots, s_K)$ be the ordered band indices used by band-selection, and X_{S} is the input image required by the pretrained patch embedding layer of the GeoFM. The selected-band image is $X_{\mathrm{sel}} = P_{\mathrm{sel}}(X)$, where the order follows the pretrained input-band slots: $X_{\mathrm{sel},j} = X_{s_j}$. This anchor preserves the standard

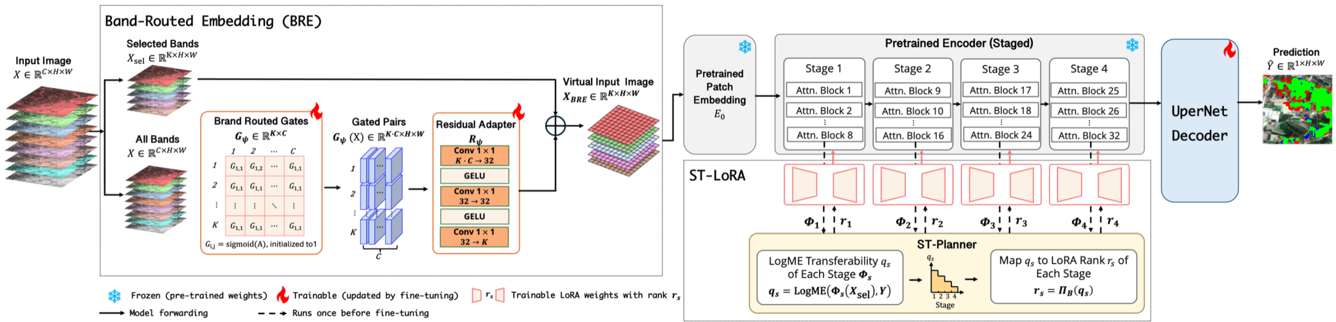

Figure 2: SPECTRA architecture. BRE maps a target image with C observed bands to a K-band pretrained-compatible virtual image by combining a selected-band path with a learned all-band residual path. ST-LoRA runs once before fine-tuning to estimate transferability using frozen encoder features and assign stage-wise LoRA ranks based. Together, they handle spectral mismatch while controlling adaptation cost.

band-selection path and keeps the input close to the distribution expected by the pretrained patch embedding.

All-band residual path. From this path, BRE learns a residual correction $\Delta_\theta(X)$ in the same K-band space to be added to the selected K bands:

$$X_{\text{BRE}} = X_{\text{sel}} + \Delta_\theta(X), \tag{4}$$

where $\Delta_\theta(X) \in \mathbb{R}^{K\times H\times W}$. The residual path is zero-initialized at its final layer, so $\Delta_\theta(X) = 0$ at model initialization. Therefore, the initial forward pass of BRE is exactly the same as band-selection: $X_{\text{BRE}} = X_{\text{sel}}$. This makes fine-tuning start from a stable selected-band baseline instead of an unconstrained learned projection.

Band routed gates. To allow every band in the fine-tuning dataset (shortened as "**fine-tuning band**" below) to contribute to every virtual pretrained band, the all-band residual path uses a learned gate table $G_\psi \in \mathbb{R}^{K\times C}$.

Each row $G_{j,:}$ corresponds to one virtual pretrained band j, and each entry $G_{j,i}$ controls the contribution of inpu band i to that band fo the virtual image (shortened as "**virtual band**" below). The gates are independent sigmoid gates:

$$G_{j,i} = 2\,\text{sigmoid}(L_{j,i}), \qquad L \in \mathbb{R}^{K\times C}. \tag{5}$$

where $L_{j,i}$ is the learnable logit for the gate connecting the fine-tuning band i to virtual band j. We initialize $L = 0$, so every gate starts from $G_{j,i} = 1$. Thus, the gates initially preserve all source-band signals rather than suppressing them.

The output of band routed gates is gated paris (as illustrated in Figure 2). Gated pairs save the contribution of observed source band c to the residual correction of virtual pretrained band k. For each virtual band j, the gated fine-tuning bands are formed by multiplying all C input channels with the corresponding gate row:

$$U_j = \text{concat}_{i=1}^{C}\left(G_{j,i}X_i\right) \in \mathbb{R}^{C\times H\times W}.$$

The gated pairs for all K virtual bands are then concatenated:

$$U = \text{concat}_{j=1}^{K} U_j \in \mathbb{R}^{KC\times H\times W}.$$

This construction ensures that no target channel is hard-discarded: every observed fine-tuning band can influence every virtual pretrained band through the residual path.

Shared residual adapter. To finally output the virtual image correction $\Delta_\theta(X)$ with K bands, concatenated gated tensor U is passed to one shared lightweight residual adapter $\Delta_\theta(X) = R_\theta(U)$, where R_θ is a compact 1×1 convolutional network as shown in Figure 2. This is a single shared adapter over the concatenated KC gated channels, not K separate target-wise adapters.

Connection to the pretrained GeoFM.. After BRE forms the virtual image X_{BRE}, the original pretrained patch embedding tokenizes it $Z = E_0(X_{\text{BRE}})$, where $Z \in \mathbb{R}^{N_p\times d}$ is the patch-token sequence with ViT encoder depth d and the number of patches after embedding [5]. Since X_{BRE} has exactly K channels, the pretrained patch embedding receives the same number of input channels as during pretraining. Thus, BRE uses all available downstream bands without modifying the pretrained patch-embedding layers.

4.2 Stage-wise Transfer-aware LoRA Rank Planning

Our proposed ST-LoRA, a stage-wise transfer-aware rank planner that runs once before fine-tuning and assigns a LoRA rank to each encoder stage. The resulting rank schedule then is fixed during training. Importantly, ST-LoRA does not route spectral bands or process image features during fine-tuning.

Let the frozen pretrained encoder be divided into S stages,

$$F_\omega = F_\omega^{(S)} \circ \cdots \circ F_\omega^{(1)}, \tag{6}$$

where ω denotes the pretrained encoder weights. To profile stage-wise transferability, ST-LoRA uses the selected-band path from BRE without applying the gated redisual path. Specifically, the selected bands are embedded by the original pretrained patch embedding:

$$Z_{\text{sel}} = E_0(X_{\text{sel}}), \qquad H_0 = Z_{\text{sel}}, \qquad H_s = F_\omega^{(s)}(H_{s-1}), \quad s = 1,\ldots,S. \tag{7}$$

For each stage s, we construct a feature matrix $\Phi_s(X_{\text{sel}})$ from the patch tokens in H_s, and then H_s goes to the next stage H_{s+1}. The class label token is removed. Patch tokens are aligned with the segmentation label map by the patch grid, and only patches whose dominant-class purity exceeds a threshold are retained. The retained patch features are flattened over the profiling images. We then measure stage-wise transferability using LogME [30], which estimates the label evidence given frozen features without fine-tuning. The transferability score of stage s is

$$q_s = \mathrm{LogME}(\Phi_s(X_{\mathrm{sel}}), Y)\,, \tag{8}$$

where Y denotes the profiling labels. The vector $q = (q_1, \ldots, q_S)$ summarizes how useful the frozen representation at each stage is for the target task. Higher values means the corresponding stage learns better for the specific downstream task than other stages.

Budget-controlled rank formulation. Now we have the transferability of each stage q_s, and the next step is to calculate the LoRA rank for each stage with limit computing cost. To control the computing cost of ST-LoRA, there needs to be an upper bound for LoRA rank assigned to each stage. For this purpose, we define a reference uniform rank r_{ref}. A uniform LoRA-r_{ref} baseline assigns $r^{\mathrm{uni}} = (r_{\mathrm{ref}}, \ldots, r_{\mathrm{ref}})$ to all S stages, so this is the standard LoRA method. We define the corresponding **total stage-rank budget** as

$$T_{\mathrm{rank}}(r_{\mathrm{ref}}) = \sum_{s=1}^{S} r_{\mathrm{ref}} = S r_{\mathrm{ref}}. \tag{9}$$

This quantity is used as an architecture-level rank budget throughout the paper. For example, with $S = 4$ stages and $r_{\mathrm{ref}} = 32$, the total rank budget matched to uniform LoRA-32 is

$$T_{\mathrm{rank}}(32) = 4 \times 32 = 128. \tag{10}$$

Thus, ST-LoRA is allowed to redistribute the same **total rank budget across** stages:

$$r_s \in \mathcal{R}(r_{\mathrm{ref}}), \qquad \sum_{s=1}^{S} r_s \le T_{\mathrm{rank}}(r_{\mathrm{ref}}). \tag{11}$$

The range of **stage-wise rank** is defined as

$$\mathcal{R}(r_{\mathrm{ref}}) = \{0, 4, 8, \ldots, r_{\mathrm{ref}}/2, r_{\mathrm{ref}}, 2r_{\mathrm{ref}}\}. \tag{12}$$

For instance,

$$\mathcal{R}(32) = \{0, 4, 8, 16, 32, 64\}. \tag{13}$$

Please note that T_{rank} is not itself a parameter count; exact trainable parameter counts are computed from the resulting schedule and reported in the experiments.

Transfer and repair planning modes. Given the transferability profile q, ST-LoRA supports two complementary planning modes. The *transfer* mode assumes that stages that are already linearly useful for the target task if it has high transferability. So it allocates more rank to stages with higher transferability scores:

$$a_s = q_s. \tag{14}$$

The *repair* mode assumes that stages with lower transferability need more corrective LoRA capacity. So it allocates more rank to stages with larger transferability gaps. The repair signal is

$$a_s = \max_s q_s - q_s. \tag{15}$$

For either mode, the stage signal a_s is combined and normalized with temperature τ:

$$w_s = \frac{\exp(a_s/\tau)}{\sum_{t=1}^{S} \exp(a_t/\tau)}. \tag{16}$$

The temperature τ controls how concentrated the allocation is across stages.

Discrete rank allocation. This step estimates stage-wise LoRA rank for both of the transfer and repair modes using score a_s with controlled rank budget. Using the budget-matched total rank budget, ST-LoRA first assigns continuous stage ranks

$$\tilde{r}_s = T_{\mathrm{rank}}(r_{\mathrm{ref}}) w_s. \tag{17}$$

Each $\tilde{r}_s$ is quantized to the largest value in $\mathcal{R}(r_{\mathrm{ref}})$ that does not exceed it:

$$r_s^{(0)} = \max\{r \in \mathcal{R}(r_{\mathrm{ref}}) : r \le \tilde{r}_s\}. \tag{18}$$

The remaining budget is then used to greedily upgrade stages when the upgrade reduces the quantization error without violating the total budget. The final stage-wise rank schedule is therefore

$$r = \Pi_{T_{\mathrm{rank}}}(q; r_{\mathrm{ref}}, \mathrm{mode}), \qquad r_s \in \mathcal{R}(r_{\mathrm{ref}}), \qquad \sum_{s=1}^{S} r_s \le T_{\mathrm{rank}}(r_{\mathrm{ref}}). \tag{19}$$

Because the constraint is an upper bound, the discretized schedule can use fewer active rank units than the uniform LoRA-r_{ref} baseline, which can reduce trainable parameters and computation while remaining budget-matched to the reference baseline.

Nested LoRA with stage-wise ranks. After planning, every adapted linear layer ℓ in stage s uses the selected rank r_s. Let W_ℓ and b_ℓ be the frozen pretrained weight and bias of layer ℓ. We use nested LoRA so that a maximum-rank adapter can be truncated to different ranks at different stages:

$$\ell(x) = W_\ell x + b_\ell + \mathbf{1}[r_s > 0]\,\frac{\alpha}{r_{\max}} B_\ell^{(r_s)} A_\ell^{(r_s)} x. \tag{20}$$

Here $r_{\max} = 2r_{\mathrm{ref}}$ is the largest rank in the range $\mathcal{R}(r_{\mathrm{ref}})$, $A_\ell^{(r_s)}$ keeps the first r_s rows of the nested LoRA matrix A_ℓ, and $B_\ell^{(r_s)}$ keeps the first r_s columns of B_ℓ. Both matrix A_ℓ, and $B_\ell^{(r_s)}$ are trainable LoRA weights defined by original LoRA paper [12]. If $r_s = 0$, the LoRA residual for that stage is disabled, and only the frozen encoder parameter exist for that stage. The pretrained encoder weights W_ℓ and b_ℓ remain frozen; only the active LoRA parameters are trained. The BRE module and the segmentation decoder are trained normally.

In our experiments, we evaluate both *transfer* and *repair* planning modes under the same reference rank r_{ref}. Unless otherwise specified, the reported SPECTRA result uses the better of the two modes selected by validation performance.

5 Experiments

5.1 Experimental Setup

Backbones and datasets. As listed in Tables 1 and 2 respectively, we evaluate three GeoFM backbones: Prithvi-EO-2.0 600M [26], ScaleMAE [24] and SatMAE [3], and four downstream datasets: Sen1Floods11 [2], FireScars, Landslide4Sense [7], and GEO-Bench South America crop-type segmentation [15]. These combinations cover binary and multiclass segmentation and expose different degrees of spectral mismatch with each backbone.

Decoder, splits, and metrics. All paper-facing runs use UPerNet [28] so that the comparison focuses on input adaptation and encoder adaptation rather than decoder choice. We use seeds 42, 43, and 44, with fixed matched settings inside each backbone-dataset cell. We

Table 4: Full main comparison across backbones and datasets. Full-FT is included as a high-cost reference, whereas LP, uniform LoRA, Surgical, and SPECTRA are PEFT methods.

Dataset	Method	Trainable params. %↓	Test macro mIoU (%)↑	Test macro F1 (%)↑	FG IoU (%)↑
GeoFM: Prithvi-EO-2.0 600M					
Sen1Floods11	LP	2.88%	85.91 ± 0.75	92.04 ± 0.49	74.95 ± 1.55
	LoRA-16	4.50%	84.46 ± 1.91	91.09 ± 1.24	72.38 ± 3.22
	LoRA-32	6.12%	85.10 ± 1.70	91.51 ± 1.09	73.52 ± 2.75
	LoRA-64	9.35%	84.40 ± 3.18	91.05 ± 2.07	72.50 ± 5.13
	Surgical	27.16%	84.81 ± 0.27	91.34 ± 0.19	73.06 ± 0.64
	Full-FT	100.00%	86.24 ± 2.95	92.23 ± 1.83	75.56 ± 5.09
	SPECTRA	4.06%	**86.39 ± 0.85**	**92.34 ± 0.53**	**75.74 ± 1.35**
FireScars	LP	2.88%	84.27 ± 0.45	91.41 ± 0.27	80.30 ± 0.66
	LoRA-16	4.50%	85.25 ± 0.35	91.99 ± 0.21	81.67 ± 0.45
	LoRA-32	6.12%	85.47 ± 0.35	92.13 ± 0.20	81.98 ± 0.41
	LoRA-64	9.35%	**85.56 ± 0.45**	**92.18 ± 0.26**	**82.10 ± 0.53**
	Surgical	27.16%	85.52 ± 0.58	92.16 ± 0.34	82.01 ± 0.71
	Full-FT	100.00%	87.71 ± 1.13	93.42 ± 0.65	84.65 ± 1.32
	SPECTRA	5.31%	85.40 ± 0.33	92.08 ± 0.19	81.75 ± 0.33
Landslide4Sense	LP	2.88%	75.33 ± 0.11	83.96 ± 0.09	43.15 ± 0.42
	LoRA-16	4.50%	76.88 ± 0.39	85.22 ± 0.32	55.31 ± 0.75
	LoRA-32	6.12%	77.07 ± 0.05	85.38 ± 0.04	53.06 ± 0.06
	LoRA-64	9.35%	77.00 ± 0.30	85.33 ± 0.23	55.62 ± 0.53
	Surgical	27.16%	76.32 ± 0.17	84.76 ± 0.14	52.40 ± 0.87
	Full-FT	100.00%	77.25 ± 0.08	85.52 ± 0.07	54.49 ± 0.40
	SPECTRA	3.29%	**77.28 ± 0.09**	**85.55 ± 0.08**	**56.11 ± 0.18**
SA Crop Type	LP	2.88%	32.86 ± 0.13	46.32 ± 0.13	31.51 ± 0.63
	LoRA-16	4.50%	35.46 ± 0.53	49.08 ± 0.59	33.27 ± 1.59
	LoRA-32	6.12%	35.61 ± 0.56	49.22 ± 0.70	34.04 ± 0.44
	LoRA-64	9.35%	35.87 ± 0.14	49.53 ± 0.26	33.42 ± 0.12
	Surgical	27.16%	34.08 ± 0.44	47.73 ± 0.51	33.08 ± 0.88
	Full-FT	100.00%	37.63 ± 1.01	51.65 ± 1.10	36.96 ± 1.58
	SPECTRA	5.31%	**36.32 ± 0.63**	**49.96 ± 0.75**	**33.58 ± 0.61**
GeoFM: ScaleMAE					
Sen1Floods11	LP	4.75%	70.16 ± 4.12	80.35 ± 3.50	48.29 ± 6.80
	LoRA-16	6.73%	76.27 ± 2.21	85.25 ± 1.71	57.92 ± 4.18
	LoRA-32	8.72%	77.07 ± 2.05	85.85 ± 1.58	59.17 ± 3.80
	LoRA-64	12.68%	77.17 ± 1.76	85.93 ± 1.33	59.34 ± 3.18
	Surgical	28.57%	73.39 ± 2.46	82.98 ± 1.99	52.98 ± 4.39
	Full-FT	100.00%	75.41 ± 0.99	84.64 ± 0.80	56.54 ± 1.93
	SPECTRA	7.23%	**85.45 ± 2.08**	**91.74 ± 1.32**	**74.11 ± 3.50**
FireScars	LP	4.75%	67.78 ± 1.12	80.49 ± 0.77	59.31 ± 0.85
	LoRA-16	6.73%	67.86 ± 1.96	80.62 ± 1.38	60.64 ± 1.60
	LoRA-32	8.72%	68.20 ± 1.89	80.86 ± 1.31	61.04 ± 1.29
	LoRA-64	12.68%	69.03 ± 0.91	81.44 ± 0.61	61.58 ± 0.40
	Surgical	28.57%	68.27 ± 1.56	80.90 ± 1.09	60.72 ± 1.31
	Full-FT	100.00%	72.30 ± 1.39	83.72 ± 0.92	65.24 ± 1.17
	SPECTRA	7.48%	**82.05 ± 2.07**	**90.05 ± 1.29**	**77.36 ± 2.94**
Landslide4Sense	LP	4.75%	74.13 ± 0.05	82.93 ± 0.04	50.18 ± 0.11
	LoRA-16	6.73%	75.05 ± 0.20	83.71 ± 0.17	51.89 ± 0.37
	LoRA-32	8.72%	75.16 ± 0.28	83.81 ± 0.23	52.12 ± 0.49
	LoRA-64	12.68%	75.15 ± 0.19	83.79 ± 0.16	52.04 ± 0.33
	Surgical	28.57%	74.82 ± 0.31	83.50 ± 0.26	51.40 ± 0.56
	Full-FT	100.00%	75.36 ± 0.07	83.97 ± 0.06	52.44 ± 0.14
	SPECTRA	6.20%	**76.33 ± 0.06**	**84.77 ± 0.06**	**54.25 ± 0.15**
SA Crop Type	LP	4.75%	30.36 ± 0.42	43.37 ± 0.71	24.44 ± 2.42
	LoRA-16	6.73%	31.50 ± 0.44	44.39 ± 0.60	26.35 ± 1.22
	LoRA-32	8.72%	31.90 ± 0.34	45.08 ± 0.58	25.91 ± 2.16
	LoRA-64	12.68%	32.09 ± 0.35	45.06 ± 0.46	26.74 ± 1.25
	Surgical	28.57%	31.03 ± 0.33	44.02 ± 0.27	25.44 ± 1.58
	Full-FT	100.00%	33.22 ± 0.52	46.58 ± 0.63	29.26 ± 1.84
	SPECTRA	7.56%	**31.97 ± 0.30**	**45.03 ± 0.36**	**29.57 ± 1.55**
GeoFM: SatMAE					
Sen1Floods11	LP	4.75%	80.83 ± 5.01	88.57 ± 3.43	66.21 ± 8.52
	LoRA-16	6.73%	83.76 ± 1.45	90.63 ± 0.94	71.05 ± 2.43
	LoRA-32	8.72%	84.95 ± 1.82	91.40 ± 1.16	73.10 ± 2.73
	LoRA-64	12.68%	84.07 ± 1.41	90.84 ± 0.90	71.61 ± 2.10
	Surgical	28.57%	82.42 ± 3.31	89.70 ± 2.23	68.77 ± 5.73
	Full-FT	100.00%	85.94 ± 2.08	92.04 ± 1.30	74.81 ± 3.35
	SPECTRA	6.49%	**87.19 ± 0.82**	**92.83 ± 0.52**	**76.85 ± 1.48**
FireScars	LP	4.75%	62.61 ± 1.00	76.82 ± 0.86	68.66 ± 1.06
	LoRA-16	6.73%	64.30 ± 0.68	78.16 ± 0.55	69.12 ± 1.16
	LoRA-32	8.72%	64.97 ± 0.30	78.67 ± 0.28	69.43 ± 1.23
	LoRA-64	12.68%	66.79 ± 0.63	80.01 ± 0.44	71.03 ± 1.42
	Surgical	28.57%	65.15 ± 2.02	78.78 ± 1.52	69.75 ± 1.52
	Full-FT	100.00%	71.61 ± 0.71	83.37 ± 0.50	76.23 ± 0.80
	SPECTRA	7.73%	**68.37 ± 1.53**	**81.10 ± 1.10**	**73.33 ± 1.35**
Landslide4Sense	LP	4.75%	76.00 ± 0.30	84.52 ± 0.25	53.76 ± 0.57
	LoRA-16	6.73%	77.46 ± 0.04	85.70 ± 0.03	56.43 ± 0.07
	LoRA-32	8.72%	77.56 ± 0.16	85.77 ± 0.14	56.61 ± 0.35
	LoRA-64	12.68%	77.58 ± 0.13	85.78 ± 0.11	56.62 ± 0.28
	Surgical	28.57%	77.37 ± 0.25	85.62 ± 0.20	56.23 ± 0.46
	Full-FT	100.00%	77.26 ± 0.13	85.53 ± 0.11	56.00 ± 0.27
	SPECTRA	6.36%	**78.26 ± 0.13**	**86.33 ± 0.11**	**57.94 ± 0.29**
SA Crop Type	LP	4.75%	30.18 ± 0.06	43.28 ± 0.17	26.69 ± 0.66
	LoRA-16	6.73%	32.42 ± 0.35	45.71 ± 0.21	29.06 ± 2.46
	LoRA-32	8.72%	32.77 ± 0.41	46.07 ± 0.39	30.29 ± 0.57
	LoRA-64	12.68%	32.69 ± 0.14	46.17 ± 0.20	29.90 ± 2.14
	Surgical	28.57%	31.20 ± 0.63	44.28 ± 0.81	28.85 ± 1.84
	Full-FT	100.00%	33.30 ± 1.12	46.54 ± 1.54	33.04 ± 2.38
	SPECTRA	7.73%	**33.39 ± 0.47**	**46.78 ± 0.49**	**28.51 ± 0.51**

report test macro mIoU, foreground IoU [18], and macro F1 [25, 27] for all segmentation tasks. All results are evaluated on the held-out test set using the best validation checkpoint selected during 50-epoch training. Parameter-efficiency comparisons report trainable parameters and trainable percentage relative to the full backbone-plus-decoder model defined as Equation 3.

Other training hyperparameters. For each GeoFM–dataset pair, we keep the loss function, learning rates, decoder architecture, and data split. The training objective combines cross-entropy [8] and Dice loss [22] for segmentation with class imbalance, with their relative weights adjusted by dynamic weight averaging (DWA) [17]. Detailed hyperparameters for all experiments are reported in our `GitHub repository`. All experiments were run on a single NVIDIA L40S GPU with 48 GB of VRAM.

Baselines. We compare SPECTRA with four fine-tuning baselines. *Linear probing* (LP) freezes the pretrained patch embedding and encoder and trains only the task-specific decoder, following the common practice of evaluating a frozen representation with a trainable readout [14]. *Uniform LoRA* freezes the pretrained backbone and trains LoRA adapters with a fixed rank $r \in \{8, 16, 32, 64\}$ in every adapted transformer block, together with the decoder [12]. For example, the method LoRA-32 applies a uniform rank of 32 throughout the pretrained encoder. *Surgical tuning* unfreezes the encoder stage selected by auto-RGN, while keeping the remaining stages frozen and training the decoder [16]. *Full fine-tuning* (**Full-FT**) trains the full pretrained backbone and the decoder, and **is treated as a high-cost reference rather than the main competitor**. Unless otherwise specified, all baselines use band-selection as the input setting. For component studies of BRE, we use LoRA-32 with band-selection as the matched reference unless otherwise stated. To test whether BRE generalizes beyond one fine-tuning policy, we also compare each baseline policy under its original band-selection input, a direct all-band MLP projector, and the same policy augmented with BRE.

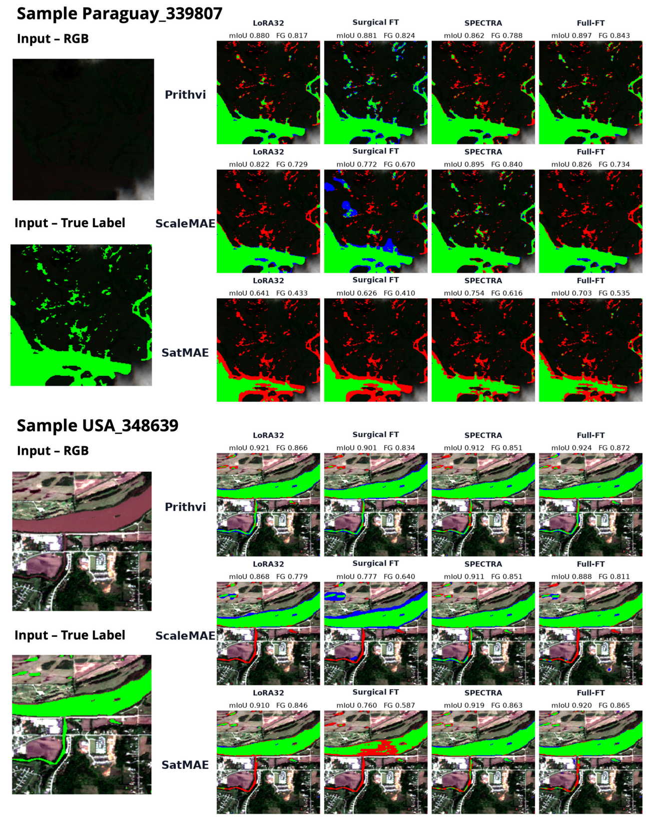

Figure 3: Visualization of predictions for LoRA-32 band-selection, surgical tuning, SPECTRA and full fine-tuning on representative binary segmentation examples. Green, blue, and red denote true positives, false positives, and false negatives, respectively.

5.2 Main Comparison

Full comparison across all backbones and datasets are in Table 4, and it is organized by GeoFM backbone and dataset. Each block compares different adaptation methods under the same dataset and GeoFM.

Overall, SPECTRA achieves strong accuracy with only 3.29–7.73% trainable parameters. It is consistently competitive with uniform LoRA and often improves over it with a smaller adaptation budget. The gains are especially clear for ScaleMAE and SatMAE, where SPECTRA is the best or near-best method on most dataset–metric pairs.

The largest improvements occur on ScaleMAE. On Sen1Floods11, SPECTRA improves macro mIoU from 77.17 with LoRA-64 and 75.41 with Full-FT to 85.45. On FireScars, it improves macro mIoU from 69.03 with LoRA-64 and 72.30 with Full-FT to 82.05. SPECTRA also exceeds Full-FT on several other settings, including Prithvi-EO-2.0 on Sen1Floods11 and Landslide4Sense, and SatMAE on Sen1Floods11, Landslide4Sense, and SA Crop Type in macro mIoU.

SPECTRA does not improve every case. For Prithvi-EO-2.0 on FireScars, uniform LoRA-64 and Full-FT remain stronger than SPECTRA. The reason might be that the pretrained bands and finetuning bands are natively matched, and the backbone is already strong for FireScars. To sum up, these results indicate that SPECTRA is not simply a universal replacement for high-capacity encoder adaptation, but a strong low-cost alternative when spectral adaptation is beneficial.

Besides the quantitative comparison, Figure 3 provides visualizations of examples. The ScaleMAE predictions show the clearest visual improvement, with SPECTRA recovering foreground regions missed by the LoRA-32 band-selection baseline. And the SPECTRA visualizations of ScaleMAE are even comparable to Prithvi, which natively learns strong features for the floods dataset. The compared baseline method Sugical FT is strong for Prithvi, but does not make good predictions when used with ScaleMAE and SatMAE. Overall speaking, SPECTRA not only enhances quantitative metrics but also makes qualitative improvements for GeoFMs that have large band mismatching with downstream tasks.

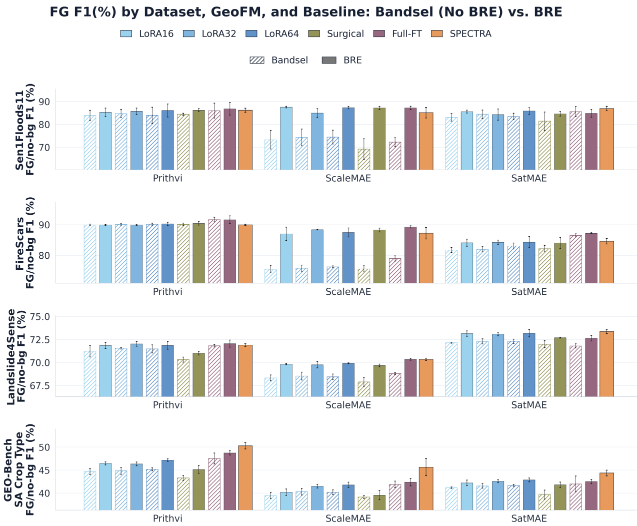

Figure 4: BRE as a general input adapter across fine-tuning policies. For each baseline policy, hatched bars use the baseline band-selection input and solid bars add BRE while keeping the same fine-tuning policy. Orange bars show the full SPECTRA pipeline with BRE and ST-LoRA.

Table 5: Baseline method with direct MLP projection and BRE. Values average macro mIoU (%) over n matched GeoFM–dataset–policy cells from three GeoFMs (Prithvi, ScaleMAE, SatMAE), four datasets (Sen1Floods11, FireScars, Landslide4Sense, and GEO-Bench SA Crop Type), and five baseline policies; deltas are relative to the same baseline policy with band-selection input.

Scope	n	band-selection	+MLP	+BRE	ΔMLP	ΔBRE
Overall	60	66.36	64.08	**69.24**	−2.28	**+2.88**
LoRA-16	12	65.89	64.26	**68.95**	−1.63	**+3.06**
LoRA-32	12	66.32	63.59	**68.90**	−2.73	**+2.59**
LoRA-64	12	66.45	63.84	**69.30**	−2.61	**+2.85**
Surgical	12	65.37	63.20	**68.84**	−2.16	**+3.48**
Full-FT	12	67.77	65.49	**70.20**	−2.28	**+2.43**

5.3 Does Extra-Band Embedding Help?

Table 5 tests whether using all available bands is beneficial and how a direct learned MLP-based projection performs compared to BRE. The table aggregates matched results over three GeoFMs, four datasets, and five baseline fine-tuning policies. The band-selection column is the same selected-band baseline summarized in Table 4, the MLP column inserts a three-layer all-band projector before the pretrained patch embedding, and the BRE column replaces only the input adapter while keeping the same baseline fine-tuning policy. Overall, BRE raises mean macro mIoU from 66.36 to 69.24 (+2.88 points), while MLP projection decreases it to 64.08 (−2.28 points). Appendix Tables 12–14 report the full matched baseline-versus-MLP rows behind the MLP column.

This comparison shows that extra bands are useful, but that the input adapter must preserve the pretrained spectral interface. A direct MLP uses all channels, but it changes the input distribution before the pretrained patch embedding and mixes selected and extra bands from a randomly initialized projection. The model therefore has to relearn a spectral tokenizer while also adapting to the downstream task. BRE avoids this failure mode: it starts exactly as band-selection through the selected-band anchor and adds a zero-initialized all-band residual path. Thus, extra bands can improve the compatible input when they are useful, while the model begins from the stable pretrained interface.

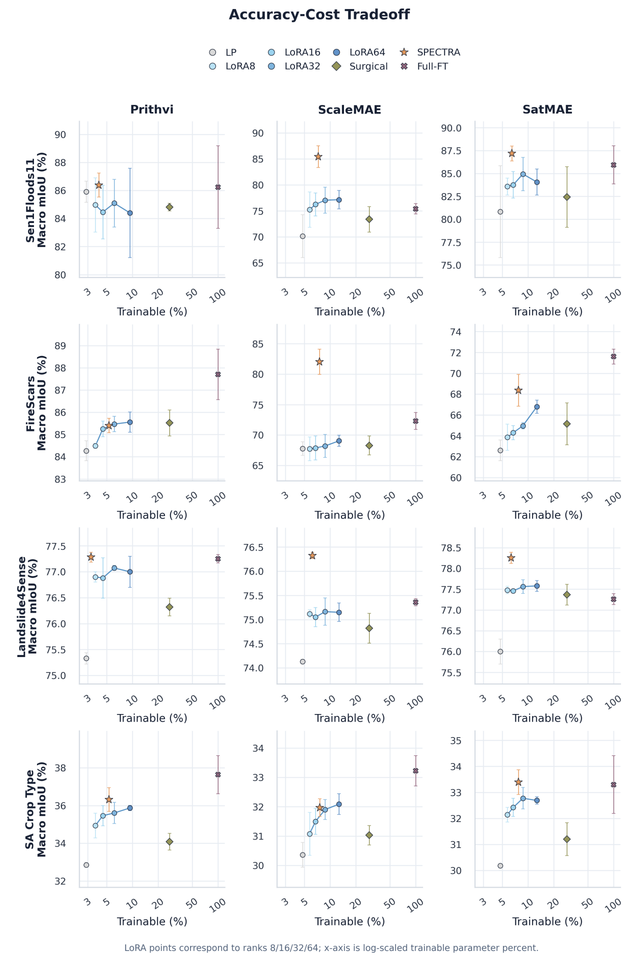

Figure 5: Accuracy-cost tradeoff curves across GeoFMs and datasets. Markers compare LP, uniform LoRA ranks, surgical tuning, SPECTRA, and full fine-tuning by trainable-parameters (%).

The broader component comparison in Figure 4 shows that this pattern is not limited to one dataset, one GeoFM, or one fine-tuning policy. For each policy, the hatched bar is the original band-selection input and the solid bar adds BRE while keeping the same encoder-adaptation policy. Across the 60 paired baseline-policy cells, BRE improves foreground/no-background F1 in 56 cells, with an average gain of +3.08 F1 points. The corresponding foreground IoU gain averages +4.15 points. The gains are positive on average for all three backbones, all four datasets, and all five baseline policies, including LoRA-16/32/64, Surgical, and Full-FT.

The largest gains occur when fixed band selection discards task-relevant spectral information. For example, Figure 4 shows large improvements for ScaleMAE on Sen1Floods11 and FireScars, where the RGB-limited pretrained interface forces band-selection to ignore many multispectral target bands. Boundary cases remain important: Prithvi on FireScars is nearly native to the HLS six-band interface, so BRE provides little benefit for that pair, and a small number of cells remain negative. These cases support a conditional interpretation: BRE is most useful when extra observed bands contain information that the selected-band baseline discards, and it is a more appropriate all-band adaptation mechanism than an unconstrained MLP projector.

5.4 Does ST-LoRA Select Useful Stage-wise Ranks?

Figure 5 and Table 6 provide complementary views of the rank-planning behavior.In Figure 5, each panel plots macro mIoU against the percentage of trainable parameters for one GeoFM–dataset pair. It compares LP, four LoRA experiments with uniform ranks, including [8, 16, 32, 64], surgical fine-tuning, SPECTRA, and full fine-tuning. Points farther left use fewer trainable parameters, and points higher are more accurate.

Table 6 isolates the ST-planner rank allocation on Sen1Floods11. Rows M1–M8 are manually designed control schedules that distribute LoRA rank across stages with different depth preferences and budgets. They use stage-wise LoRA but **without our automatic ST-planner**. A rank vector $[r_1, r_2, r_3, r_4]$ gives the stage-wise LoRA rank assigned from shallow to deep encoder stages, and the rank budget is $\sum_s r_s$. The row "SPECTRA" uses ST-planner to plan the rank $[r_1, r_2, r_3, r_4]$. For the final SPECTRA configuration in Table 4, Prithvi with Sen1Floods11 uses ST-LoRA Repair and reaches 86.39±0.85 macro mIoU, +0.43 percentage points above the uniform LoRA-32 BRE reference while using fewer active LoRA parameters. For ScaleMAE, the strongest automatic BRE schedule is ST-LoRA Transfer with rank vector $[16, 32, 16, 16]$. It reaches 85.45 ± 2.08 macro mIoU, slightly above the uniform LoRA-32 BRE reference. In Figure 5, these SPECTRA points generally sit far left of full fine-tuning and surgical fine-tuning, while remaining competitive with or above nearby LoRA ranks.

These results support the efficiency claim, and they also show that a larger rank budget does not automatically lead to better performance. Manual schedules remain useful controls when sufficient time and computing resources are available to test different rank allocations. The advantage of ST-LoRA is that it automatically finds an efficient stage-wise rank schedule without exhaustively searching over the $|\mathcal{R}|^S$ possible combinations of $[r_1, r_2, r_3, r_4]$.

Table 6: Comparison between ST-LoRA and manually-selected ranks on Sen1Floods11.

Method & ranks	Rank budget	Trainable params	Macro mIoU %
GeoFM: Prithvi-EO-2.0 600M			
M1 [4,4,8,16]	32	23.94M	**86.58 ± 0.71**
M2 [16,8,4,4]	32	23.94M	84.98 ± 2.37
M3 [8,16,16,8]	48	26.57M	85.85 ± 2.13
M4 [4,8,16,32]	60	28.53M	85.85 ± 1.47
M5 [32,16,16,8]	72	30.50M	85.02 ± 2.39
M6 [8,16,24,32]	80	31.81M	85.22 ± 1.89
M7 [4,8,16,64]	92	33.77M	86.05 ± 1.05
M8 [64,32,16,8]	120	38.36M	85.15 ± 2.89
ST-LoRA [8,16,16,8]	**48**	**26.35M**	85.88 ± 1.24
GeoFM: ScaleMAE			
M1 [4,4,8,16]	**32**	**18.23M**	**85.27 ± 1.77**
M2 [16,8,4,4]	32	18.23M	84.98 ± 1.82
M3 [8,16,16,8]	48	19.80M	84.63 ± 1.75
M4 [4,8,16,32]	60	20.98M	85.15 ± 1.83
M5 [32,16,16,8]	72	22.16M	84.99 ± 1.67
M6 [8,16,24,32]	80	22.95M	85.25 ± 1.68
M7 [4,8,16,64]	92	24.13M	85.12 ± 1.89
M8 [64,32,16,8]	120	26.88M	85.01 ± 1.59
ST-LoRA [16,32,16,16]	80	22.95M	**85.29 ± 2.17**

6 Discussion and Limitations

When should extra-band adapters be used? The practical rule from the experiments is conditional. If spectral mismatch is low and band-selection is already strong, the safest default may be to keep band-selection. The Prithvi–FireScars result illustrates this boundary condition: the pretrained and fine-tuning inputs share the same six HLS spectral bands, so there are no discarded channels for BRE to recover, and the grouped-bar results show no performance enhancement. If many target bands are discarded or the target sensor contains task-relevant channels outside the selected subset, BRE should be tested under matched training settings. The largest observed gains are consistent with this rule, especially for ScaleMAE on multispectral downstream datasets.

Limitations. The current study uses constrained hyperparameters and a common UPerNet decoder to control variables across backbones and datasets. This is appropriate for method diagnosis, but it is not a claim of published state-of-the-art performance on every dataset. BRE gains are not universal, and their magnitude can change by backbone and task. The ST-LoRA evidence is strongest on the completed Sen1Floods11 rank-planner runs; broader rank-planner validation across all target datasets remains necessary before claiming universal parameter-efficiency gains.

7 Conclusion

We introduced SPECTRA, a parameter-efficient fine-tuning framework for cross-sensor GeoFM fine-tuning under spectral mismatch. The proposed BRE routes all available target bands into a pretrained-compatible virtual input, while ST-LoRA allocates LoRA rank across encoder stages using transferability diagnostics. SPECTRA provides a favorable accuracy–cost tradeoff. The results reveal that extra-band embedding helps most when band-selection discards useful channels, and stage-wise rank planning can reduce LoRA budget while preserving competitive accuracy.

References

[1] Guillaume Astruc, Nicolas Gonthier, Clément Mallet, and Loïc Landrieu. 2025. AnySat: One Earth Observation Model for Many Resolutions, Scales, and Modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19530–19540. doi:10.1109/CVPR52734.2025.01819

[2] Derrick Bonafilia, Beth Tellman, Tyler Anderson, and Erica Issenberg. 2020. Sen1Floods11: A Georeferenced Dataset to Train and Test Deep Learning Flood Algorithms for Sentinel-1. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 835–845.

[3] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. 2022. SatMAE: Pre-training Transformers for Temporal and Multi-Spectral Satellite Imagery. In *Advances in Neural Information Processing Systems*, Vol. 35. 197–211.

[4] Zhe Dong, Yanfeng Gu, and Tianzhu Liu. 2024. UPetu: A Unified Parameter-Efficient Fine-Tuning Framework for Remote Sensing Foundation Model. *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), 1–13. doi:10.1109/TGRS.2024.3382734

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[7] Omid Ghorbanzadeh, Yonghao Xu, Pedram Ghamisi, Michael Kopp, and David Kreil. 2022. Landslide4Sense: Reference Benchmark Data and Deep Learning Models for Landslide Detection. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–17. doi:10.1109/TGRS.2022.3215209

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[9] Boran Han, Shuai Zhang, Xingjian Shi, and Markus Reichstein. 2024. Bridging Remote Sensors with Multisensor Geospatial Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 27852–27862. doi:10.1109/CVPR52733.2024.02631

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16000–16009.

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2790–2799.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[13] Johannes Jakubik, Sujit Roy, Christopher E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, et al. 2023. Foundation Models for Generalist Geospatial Artificial Intelligence. arXiv:2310.18660 [cs.CV] https://arxiv.org/abs/2310.18660

[14] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. 2019. Do Better ImageNet Models Transfer Better?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2661–2671.

[15] Alexandre Lacoste, Nils Lehmann, Pau Rodriguez, Evan Sherwin, Hannah Kerner, Bj"orn L"utjens, Jeremy Irvin, David Dao, Hamed Alemohammad, Alexandre Drouin, Mehmet Gunturkun, Gabriel Huang, David Vazquez, Dava Newman, Yoshua Bengio, Stefano Ermon, and Xiaoxiang Zhu. 2023. GEO-Bench: Toward Foundation Models for Earth Monitoring. In *Advances in Neural Information Processing Systems*, Vol. 36.

[16] Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2023. Surgical Fine-Tuning Improves Adaptation to Distribution Shifts. In *International Conference on Learning Representations*.

[17] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1871–1880.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[19] Gengchen Mai, Weiming Huang, Jin Sun, Shenzhe Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, Chris Cundy, Ziyuan Li, Rui Zhu, and Ni Lao. 2024. On the Opportunities and Challenges of Foundation Models for GeoAI. *ACM Transactions on Spatial Algorithms and Systems* 10, 2, Article 11 (2024), 46 pages. doi:10.1145/3653070

[20] Valerio Marsocci, Yuru Jia, Georges Le Bellier, David Kerekes, Liang Zeng, Sebastian Hafner, Sebastian Gerard, Eric Brune, Ritu Yadav, Ali Shibli, Heng Fang, Yifang Ban, Maarten Vergauwen, Nicolas Audebert, and Andrea Nascetti. 2026. PANGAEA: Assessing Geospatial Foundation Models Capabilities through a Global and Inclusive Benchmark. *IEEE Geoscience and Remote Sensing Magazine* 14, 1 (2026), 245–285. doi:10.1109/MGRS.2025.3628194

[21] Francesc Marti Escofet, Benedikt Blumenstiel, Linus Scheibenreif, Paolo Fraccaro, and Konrad Schindler. 2025. Fine-tune smarter, not harder: Parameter-efficient fine-tuning for geospatial foundation models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 516–532.

[22] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *International Conference on 3D Vision*. 565–571.

[23] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. 2020. LEEP: A New Measure to Evaluate Transferability of Learned Representations. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research)*.

[24] Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. 2023. Scale-MAE: A Scale-Aware Masked Autoencoder for Multiscale Geospatial Representation Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4088–4099.

[25] Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45, 4 (2009), 427–437.

[26] Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, Orsteinn Eli Gislason, Benedikt Blumenstiel, Rinki Ghosal, Pedro Henrique de Oliveira, Joao Lucas de Sousa Almeida, Rocco Sedona, Yanghui Kang, Srija Chakraborty, Sizhe Wang, Carlos Gomes, Ankur Kumar, Vishal Gaur, Myscon Truong, Denys Godwin, Sam Khallaghi, Hyunho Lee, Chia Yu Hsu, Ata Akbari Asanjan, Besart Mujeci, Disha Shidham, Rufai Omowunmi Balogun, Venkatesh Kolluru, Trevor Keenan, Paulo Arevalo, Wenwen Li, Hamed Alemohammad, Pontus Olofsson, Timothy Mayer, Christopher Hain, Robert Kennedy, Bianca Zadrozny, David Bell, Gabriele Cavallaro, Campbell Watson, Manil Maskey, Rahul Ramachandran, and Juan Bernabe Moreno. 2025. Prithvi-EO-2.0: A Versatile Multi-Temporal Foundation Model for Earth Observation Applications. *IEEE Transactions on Geoscience and Remote Sensing* (2025). doi:10.1109/TGRS.2025.3642610

[27] Cornelis Joost van Rijsbergen. 1979. *Information Retrieval* (2 ed.). Butterworth-Heinemann.

[28] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. 2018. Unified Perceptual Parsing for Scene Understanding. In *Proceedings of the European Conference on Computer Vision*. 418–434.

[29] Zhitong Xiong, Yi Wang, Fahong Zhang, Adam J. Stewart, Jo"elle Hanna, Damian Borth, Ioannis Papoutsis, Bertrand Le Saux, Gustau Camps-Valls, and Xiao Xiang Zhu. 2024. Neural Plasticity-Inspired Multimodal Foundation Model for Earth Observation. arXiv:2403.15356 [cs.CV] https://arxiv.org/abs/2403.15356

[30] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. LogME: Practical Assessment of Pre-trained Models for Transfer Learning. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 12133–12143. https://proceedings.mlr.press/v139/you21b.html

A Spectral Interface and Band-Selection Diagnostics

This appendix lists the spectral metadata and band-selection mappings behind Table 3.

Table 7: Pretrained spectral interfaces.

Backbone/source	Index	Band name	Center (nm)	FWHM (nm)
Prithvi pretraining	0	BLUE	490.0	65.0
	1	GREEN	560.0	35.0
	2	RED	665.0	30.0
	3	NIR_NARROW	865.0	20.0
	4	SWIR_1	1610.0	90.0
	5	SWIR_2	2202.0	180.0
ScaleMAE pretraining	0	RED	665.0	80.0
	1	GREEN	560.0	80.0
	2	BLUE	490.0	80.0
SatMAE pretraining	0	B2_BLUE	490.0	65.0
	1	B3_GREEN	560.0	35.0
	2	B4_RED	665.0	30.0
	3	B5_RE1	705.0	15.0
	4	B6_RE2	740.0	15.0
	5	B7_RE3	783.0	20.0
	6	B8_NIR	842.0	115.0
	7	B8A_RE4	865.0	20.0
	8	B11_SWIR1	1610.0	90.0
	9	B12_SWIR2	2202.0	180.0

Table 8: Downstream target-channel metadata.

Dataset	Index	Band name	Center (nm)	FWHM (nm)
Sen1Floods11	0	B1_COASTAL	443.0	20.0
	1	B2_BLUE	490.0	65.0
	2	B3_GREEN	560.0	35.0
	3	B4_RED	665.0	30.0
	4	B5_RE1	705.0	15.0
	5	B6_RE2	740.0	15.0
	6	B7_RE3	783.0	20.0
	7	B8_NIR	842.0	115.0
	8	B8A_RE4	865.0	20.0
	9	B9_WATER_VAPOR	945.0	20.0
	10	B10_CIRRUS	1375.0	30.0
	11	B11_SWIR1	1610.0	90.0
	12	B12_SWIR2	2202.0	180.0
FireScars	0	BLUE	490.0	65.0
	1	GREEN	560.0	35.0
	2	RED	665.0	30.0
	3	NIR_NARROW	865.0	20.0
	4	SWIR_1	1610.0	90.0
	5	SWIR_2	2202.0	180.0
Landslide4Sense	0	B1_COASTAL	443.0	20.0
	1	B2_BLUE	490.0	65.0
	2	B3_GREEN	560.0	35.0
	3	B4_RED	665.0	30.0
	4	B5_RE1	705.0	15.0
	5	B6_RE2	740.0	15.0
	6	B7_RE3	783.0	20.0
	7	B8_NIR	842.0	115.0
	8	B8A_RE4	865.0	20.0
	9	B9_WATER_VAPOR	945.0	20.0
	10	B10_CIRRUS	1375.0	30.0
	11	B11_SWIR1	1610.0	90.0
	12	B12_SWIR2	2202.0	180.0
	13	AUX	0.0	0.0
SA Crop Type	0	B1_COASTAL	443.0	20.0
	1	B2_BLUE	490.0	65.0
	2	B3_GREEN	560.0	35.0
	3	B4_RED	665.0	30.0
	4	B5_RE1	705.0	15.0
	5	B6_RE2	740.0	15.0
	6	B7_RE3	783.0	20.0
	7	B8_NIR	842.0	115.0
	8	B8A_RE4	865.0	20.0
	9	B9_WATER_VAPOR	945.0	20.0
	10	B11_SWIR1	1610.0	90.0
	11	B12_SWIR2	2202.0	180.0

Table 9: Prithvi band-selection source-to-target mapping.

Dataset	Target idx	Target band	Status	Matched native band(s)
Sen1Floods11	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	selected	NIR_NARROW
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	selected	SWIR_1
	12	B12_SWIR2	selected	SWIR_2
FireScars	0	BLUE	selected	BLUE
	1	GREEN	selected	GREEN
	2	RED	selected	RED
	3	NIR_NARROW	selected	NIR_NARROW
	4	SWIR_1	selected	SWIR_1
	5	SWIR_2	selected	SWIR_2
Landslide4Sense	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	selected	NIR_NARROW
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	selected	SWIR_1
	12	B12_SWIR2	selected	SWIR_2
	13	AUX	auxiliary	–
SA Crop Type	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	selected	NIR_NARROW
	9	B9_WATER_VAPOR	extra	–
	10	B11_SWIR1	selected	SWIR_1
	11	B12_SWIR2	selected	SWIR_2

Table 10: ScaleMAE band-selection source-to-target mapping.

Dataset	Target idx	Target band	Status	Matched native band(s)
Sen1Floods11	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	extra	–
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	extra	–
	12	B12_SWIR2	extra	–
FireScars	0	BLUE	selected	BLUE
	1	GREEN	selected	GREEN
	2	RED	selected	RED
	3	NIR_NARROW	extra	–
	4	SWIR_1	extra	–
	5	SWIR_2	extra	–
Landslide4Sense	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	extra	–
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	extra	–
	12	B12_SWIR2	extra	–
	13	AUX	auxiliary	–
SA Crop Type	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	BLUE
	2	B3_GREEN	selected	GREEN
	3	B4_RED	selected	RED
	4	B5_RE1	extra	–
	5	B6_RE2	extra	–
	6	B7_RE3	extra	–
	7	B8_NIR	extra	–
	8	B8A_RE4	extra	–
	9	B9_WATER_VAPOR	extra	–
	10	B11_SWIR1	extra	–
	11	B12_SWIR2	extra	–

Table 11: SatMAE band-selection source-to-target mapping.

Dataset	Target idx	Target band	Status	Matched native band(s)
Sen1Floods11	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	B2_BLUE
	2	B3_GREEN	selected	B3_GREEN
	3	B4_RED	selected	B4_RED
	4	B5_RE1	selected	B5_RE1
	5	B6_RE2	selected	B6_RE2
	6	B7_RE3	selected	B7_RE3
	7	B8_NIR	selected	B8_NIR
	8	B8A_RE4	selected	B8A_RE4
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	selected	B11_SWIR1
	12	B12_SWIR2	selected	B12_SWIR2
FireScars	0	BLUE	selected	B2_BLUE
	1	GREEN	selected	B3_GREEN
	2	RED	selected	B4_RED; B5_RE1; B6_RE2
	3	NIR_NARROW	selected	B7_RE3; B8_NIR; B8A_RE4
	4	SWIR_1	selected	B11_SWIR1
	5	SWIR_2	selected	B12_SWIR2
Landslide4Sense	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	B2_BLUE
	2	B3_GREEN	selected	B3_GREEN
	3	B4_RED	selected	B4_RED
	4	B5_RE1	selected	B5_RE1
	5	B6_RE2	selected	B6_RE2
	6	B7_RE3	selected	B7_RE3
	7	B8_NIR	selected	B8_NIR
	8	B8A_RE4	selected	B8A_RE4
	9	B9_WATER_VAPOR	extra	–
	10	B10_CIRRUS	extra	–
	11	B11_SWIR1	selected	B11_SWIR1
	12	B12_SWIR2	selected	B12_SWIR2
	13	AUX	auxiliary	–
SA Crop Type	0	B1_COASTAL	extra	–
	1	B2_BLUE	selected	B2_BLUE
	2	B3_GREEN	selected	B3_GREEN
	3	B4_RED	selected	B4_RED
	4	B5_RE1	selected	B5_RE1
	5	B6_RE2	selected	B6_RE2
	6	B7_RE3	selected	B7_RE3
	7	B8_NIR	selected	B8_NIR
	8	B8A_RE4	selected	B8A_RE4
	9	B9_WATER_VAPOR	extra	–
	10	B11_SWIR1	selected	B11_SWIR1
	11	B12_SWIR2	selected	B12_SWIR2

B Full Direct-MLP Companion Results

Tables 12–14 expand the direct-MLP companion experiments summarized in Table 5. Each row keeps the GeoFM, dataset, and baseline fine-tuning policy fixed, and changes only the input adapter from band-selection to a direct all-band MLP projection. These results do not include ST-LoRA. Values are seed mean ± sample standard deviation test macro mIoU (%) over seeds 42–44; Δ is +MLP minus the matched band-selection baseline.

Table 12: Direct-MLP companions for Prithvi.

Dataset	Policy	band-selection	+MLP	Δ
Sen1Floods11	LoRA-16	84.46±1.91	84.24±2.00	−0.22
Sen1Floods11	LoRA-32	85.10±1.70	83.53±2.11	−1.57
Sen1Floods11	LoRA-64	84.40±3.18	84.14±2.26	−0.26
Sen1Floods11	Surgical	84.81±0.27	83.40±1.99	−1.41
Sen1Floods11	Full-FT	86.24±2.95	86.25±2.12	+0.00
FireScars	LoRA-16	85.25±0.35	85.05±0.68	−0.20
FireScars	LoRA-32	85.47±0.35	85.03±0.78	−0.43
FireScars	LoRA-64	85.56±0.45	85.78±1.46	+0.21
FireScars	Surgical	85.52±0.58	83.37±1.71	−2.16
FireScars	Full-FT	87.71±1.13	85.31±0.72	−2.40
Landslide4Sense	LoRA-16	76.88±0.39	76.16±0.51	−0.73
Landslide4Sense	LoRA-32	77.07±0.05	75.88±1.31	−1.20
Landslide4Sense	LoRA-64	77.00±0.30	76.30±0.36	−0.70
Landslide4Sense	Surgical	76.32±0.17	75.91±0.05	−0.41
Landslide4Sense	Full-FT	77.25±0.08	76.37±0.50	−0.88
SA Crop Type	LoRA-16	35.46±0.53	32.57±0.17	−2.89
SA Crop Type	LoRA-32	35.61±0.56	33.18±0.40	−2.43
SA Crop Type	LoRA-64	35.87±0.14	33.09±0.27	−2.78
SA Crop Type	Surgical	34.08±0.44	30.98±0.41	−3.10
SA Crop Type	Full-FT	37.63±1.01	34.67±0.23	−2.97

Table 13: Direct-MLP companions for ScaleMAE.

Dataset	Policy	band-selection	+MLP	Δ
Sen1Floods11	LoRA-16	76.27±2.71	81.52±2.38	+5.25
Sen1Floods11	LoRA-32	77.07±2.51	79.84±2.35	+2.77
Sen1Floods11	LoRA-64	77.17±2.16	79.92±3.96	+2.75
Sen1Floods11	Surgical	73.39±3.01	79.92±0.58	+6.54
Sen1Floods11	Full-FT	75.41±1.22	83.76±1.76	+8.35
FireScars	LoRA-16	67.86±1.96	64.29±4.60	−3.57
FireScars	LoRA-32	68.20±1.89	55.30±20.09	−12.90
FireScars	LoRA-64	69.03±0.91	57.60±21.83	−11.43
FireScars	Surgical	68.27±1.56	60.72±7.67	−7.55
FireScars	Full-FT	72.30±1.39	67.97±16.76	−4.33
Landslide4Sense	LoRA-16	75.05±0.20	73.30±2.96	−1.75
Landslide4Sense	LoRA-32	75.16±0.28	72.63±2.77	−2.54
Landslide4Sense	LoRA-64	75.15±0.19	72.63±4.12	−2.52
Landslide4Sense	Surgical	74.82±0.31	68.82±9.41	−6.00
Landslide4Sense	Full-FT	75.36±0.07	74.16±0.60	−1.20
SA Crop Type	LoRA-16	31.50±0.44	30.92±1.09	−0.58
SA Crop Type	LoRA-32	31.90±0.34	30.64±1.40	−1.27
SA Crop Type	LoRA-64	32.09±0.35	30.46±1.94	−1.62
SA Crop Type	Surgical	31.03±0.33	30.50±1.53	−0.53
SA Crop Type	Full-FT	33.22±0.52	27.63±7.95	−5.59

Table 14: Direct-MLP companions for SatMAE.

Dataset	Policy	band-selection	+MLP	Δ
Sen1Floods11	LoRA-16	83.76±1.45	83.48±1.30	−0.28
Sen1Floods11	LoRA-32	84.95±1.82	84.07±0.43	−0.87
Sen1Floods11	LoRA-64	84.07±1.41	84.81±0.12	+0.74
Sen1Floods11	Surgical	82.42±3.31	83.05±2.79	+0.63
Sen1Floods11	Full-FT	85.94±2.08	85.22±0.56	−0.73
FireScars	LoRA-16	64.30±0.68	58.31±4.75	−5.99
FireScars	LoRA-32	64.97±0.30	61.08±5.41	−3.88
FireScars	LoRA-64	66.79±0.63	59.98±7.61	−6.81
FireScars	Surgical	65.15±2.02	59.67±7.30	−5.48
FireScars	Full-FT	71.61±0.71	59.00±4.19	−12.61
Landslide4Sense	LoRA-16	77.46±0.04	75.43±0.79	−2.04
Landslide4Sense	LoRA-32	77.56±0.16	76.08±0.86	−1.48
Landslide4Sense	LoRA-64	77.58±0.13	76.00±0.19	−1.58
Landslide4Sense	Surgical	77.37±0.25	75.19±0.75	−2.19
Landslide4Sense	Full-FT	77.26±0.13	75.46±0.91	−1.80
SA Crop Type	LoRA-16	32.42±0.35	25.89±1.12	−6.53
SA Crop Type	LoRA-32	32.77±0.41	25.82±0.90	−6.95
SA Crop Type	LoRA-64	32.69±0.14	25.33±3.95	−7.35
SA Crop Type	Surgical	31.20±0.63	26.88±0.20	−4.32
SA Crop Type	Full-FT	33.30±1.12	30.12±0.20	−3.18