GeoSANE: Learning Geospatial Representations from Models, Not Data

Joëlle Hanna[1] Damian Falk[1] Stella X. Yu[2] Damian Borth[1,3]
[1] University of St.Gallen [2] University of Michigan and UC Berkeley [3] ESA Φ-Lab

Abstract

Recent advances in remote sensing have led to an increase in the number of available foundation models; each trained on different modalities, datasets, and objectives, yet capturing only part of the vast geospatial knowledge landscape. While these models show strong results within their respective domains, their capabilities remain complementary rather than unified. Therefore, instead of choosing one model over another, we aim to combine their strengths into a single shared representation. We introduce GeoSANE, a geospatial model foundry that learns a unified neural representation from the weights of existing foundation models and task-specific models, able to generate novel neural networks weights on-demand. Given a target architecture, GeoSANE generates weights ready for finetuning for classification, segmentation, and detection tasks across multiple modalities. Models generated by GeoSANE consistently outperform their counterparts trained from scratch, match or surpass state-of-the-art remote sensing foundation models, and outperform models obtained through pruning or knowledge distillation when generating lightweight networks. Evaluations across ten diverse datasets and on GEO-Bench confirm its strong generalization capabilities. By shifting from pre-training to weight generation, GeoSANE introduces a new framework for unifying and transferring geospatial knowledge across models and tasks. Code and data are available at github.com/HSG-AIML/GeoSANE.

1. Introduction

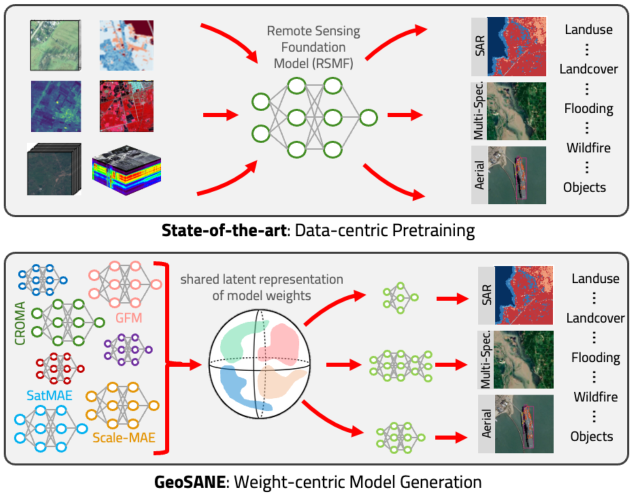

Figure 1. Instead of pretraining geospatial (foundation) models from satellite data (top), GeoSANE proposes to leverage publicly available geospatial (foundation) models to train a shared latent representation encapsulating their knowledge to generate model weights tailored for specific downstream task (bottom).

Foundation models have transformed computer vision and remote sensing by providing strong, task-agnostic representations capable of adapting towards a diverse set of downstream tasks. In remote sensing, foundation models [1, 7, 8, 12, 14, 19, 20, 31, 43, 51] have demonstrated that large-scale pretraining on multispectral and multimodal satellite data yields transferable representations for classification, segmentation, and detection. However, despite their success, the landscape of remote sensing foundation models (RSFMs) remains fragmented: each model specializes in a subset of sensors, spatial resolutions, or objectives, with many models being complementary and some being more comprehensive than others. Recent surveys [24, 50] count more than 70 RSFMs, with more RSFM joining this list. As a result, users must repeatedly decide which RSFM to select or retrain for a new task, despite the fact that the union of all these models probably encapsulates a broader range of geospatial knowledge than any single RSFM alone.

In this work, we propose to take a fundamentally different perspective. Instead of learning yet another foundation model from remote sensing data, we propose to learn from existing models themselves — directly from their parameters in weight space. Inspired by recent advances in *weight space learning* [28, 35, 36, 38, 39, 46], we treat the weights of trained neural networks as input modality to learn a single shared latent representation of a given population of neural network models [37]. This approach would allow us to combine the knowledge encoded in multiple pretrained neural networks such that one could efficiently generate new model weights, being more suitable for task-specific fine-tuning but without the cost of large-scale pretraining.

We introduce GeoSANE, a sequential autoencoder for neural embeddings acting as a geospatial model foundry that learns a shared latent representation across diverse RSFMs and task-specific remote sensing models. GeoSANE leverages a transformer-based encoder-decoder in weight space to embed these heterogeneous neural networks spanning different architectures, sensing modalities, and objectives, into a shared latent manifold. From this manifold, GeoSANE can sample new weights to generate entire models for a target architectures and a given remote sensing downstream task. In doing so, GeoSANE shifts the paradigm from *data-centric pretraining* to *weight-centric generation* of remote sensing models (Fig. 1)

The proposed approach is motivated by three key points: (i) The availability of open-source geospatial models on platforms such as Hugging Face provides a rich collection of pretrained models capturing a vast amount of domain-specific knowledge. (ii) Operating directly in weight space decouples knowledge transfer from data availability, enabling efficient adaptation across sensors, modalities, and tasks. (iii) Pretrained models constrain their task-specific models to be of similar architecture and size demanding an additional distillation step to build lightweight models. GeoSANE aims to address all the points above, as it collects a heterogeneous population of existing geospatial models, tokenizes their parameters, and learns a shared latent embedding across them. Given a target architecture (being small or large), GeoSANE samples from this latent space to produce functional models whose weights encode the aggregated knowledge of the full population.

We evaluate GeoSANE on 10 remote sensing datasets spanning optical, multispectral, and radar modalities, and across classification, segmentation, and detection tasks. Generated models consistently outperform training from scratch, match or exceed leading foundation models, and outperform pruning and distillation baselines when used to create lightweight neural networks. These results demonstrate strong generalization and the feasibility of model generation in weight space for geospatial domains. In summary, our contributions are threefold:

- A new paradigm for remote sensing pretraining. We propose to learn from existing geospatial models in weight space rather than from remote sensing data.
- GeoSANE, a scalable encoder-decoder approach that trains from heterogeneous models and enables on-demand weight generation for arbitrary architectures, modalities, and tasks.
- Comprehensive empirical validation across multiple tasks, sensors, and datasets, establishing weight space learning as a viable alternative to regular pretraining.

Together, these results indicate that learning from existing models in weight space provides a novel path to harness the rich landscape of geospatial foundation models.

2. Related Work

Remote Sensing Foundation Models. Recent progress in remote sensing foundation models (RSFMs) enabled the reuse of pretrained models for many different satellite image analysis tasks. Early models such as SeCo [25], SSL4EO-S12 [47], self-supervised ViT [33] and SatMAE [7] demonstrated the benefits of large-scale pretraining on Sentinel imagery [9, 44] for downstream classification and segmentation. Later works have greatly expanded model capacity and data diversity. Scale-MAE [31] introduced scale-aware pretraining for multi-resolution data, while Prithvi-EO [19] and its successor Prithvi-EO-2.0 [43] leveraged multimodal Landsat-Sentinel fusion to build general-purpose geospatial transformers. More recent foundation models have grown larger and more versatile, covering more sensors, data types, and tasks. CROMA [12] proposed cross-modal alignment between optical and radar inputs to improve multimodal understanding. DOFA [51] introduced domain-oriented pretraining for aerial imagery, emphasizing generalization across data sources. RingMo [42] explored large-scale multimodal transformers trained on diverse global datasets, while AnySat [1] aimed for universality across sensors and resolutions. Building on these trends, TerraFM [8] scaled model and data size to a continental level with multisensor pretraining, and TerraMind [20] extended this direction toward generative, multimodal foundation models.

In contrast, GeoSANE moves beyond traditional satellite imagery based pretraining by learning directly from existing remote sensing model weights to combine their encoded knowledge in an shared latent representation.

Model Weight Generation. Early work on parameter generation used hypernetworks [15], where a separate network predicts the weights of a target model. More recent work instead treats trained weights as a data modality, learning directly from weight space. Hyper-Representations [35] showed that one can learn a lower-dimensional manifold from a population of neural network models and that this shared latent representation can be exploited to generate functional models [36, 38]. Recent work further demonstrated that such representations can be learned from heterogeneous models hosted on hubs [11], removing the need for curated model zoos. More recent diffusion-based approaches, including G.pt [28], Neural Network Diffusion [46] and D2NWG [39], model the distribution of trained weights to generate parameters conditioned on dataset or architecture, demonstrating the feasibility of sampling performant networks directly in weight space.

Despite these advances, learning from weights mostly remains limited to either homogeneous model populations [28, 36, 38, 39, 46] or traditional computer vision models [11]. This work addresses, for the first time, learn-

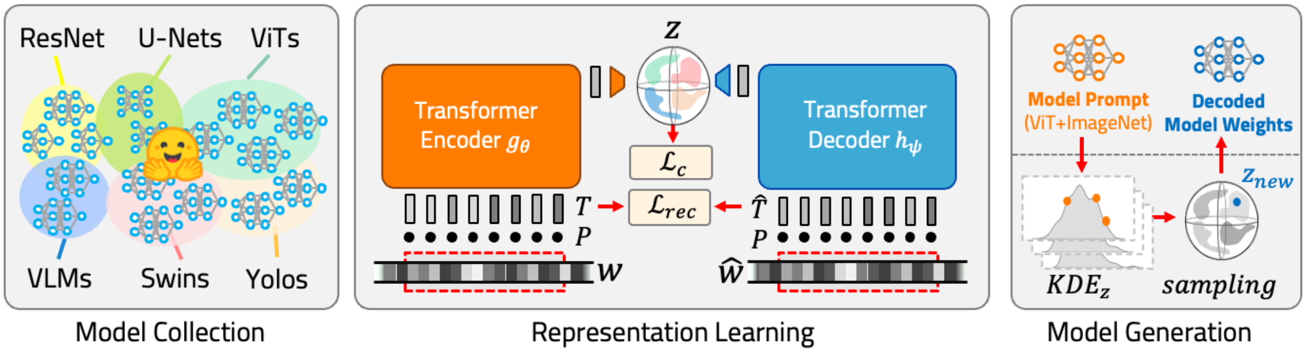

Figure 2. Overview of our approach. (Left) A heterogeneous collection of models, including ViTs, Swins, ResNets, UNets, and vision-language models, is gathered from HuggingFace. (Center) A weight-space autoencoder is trained to reconstruct and embed these models into a shared latent representation. (Right) From this latent space, GeoSANE can generate new models on demand for specific downstream tasks such as flood segmentation, object detection, or land-cover classification.

ing from weights of fundamentally different architectures processing non-RGB based input modalities and goes beyond image classification to pixel-wise segmentation and object detection.

Model Merging. Model merging enables the combination of multiple models into a single one. Early work focused on weight-space ensembling, where models fine-tuned from a shared initialization are combined by averaging or interpolating their parameters, as demonstrated in Model Soups [49] and WiSE-FT [48]. However, when the merged models have been fine-tuned on divergent tasks, simple averaging often leads to performance degradation due to conflicting parameter updates. To address these limitations, recent methods like TIES-Merging [52] prune insignificant weight changes and align important updates before averaging, while DARE [53] zeroes small deltas and amplifies larger ones to reduce interference. Nonetheless, most of these techniques assume that all models share the same architecture and initialization.

In contrast, GeoSANE is designed to encapsulate the knowledge of a large collection of remote sensing models, regardless of their architecture or initialization, overcoming key limitations of prior model merging methods.

3. Method

GeoSANE works in three stages (Fig. 2). First, we collect a heterogeneous set of remote sensing models that cover different architectures, tasks, and sensing modalities. Next, we train a weight-space autoencoder to embed these models into a shared latent representation. Finally, given a user defined prompt model, we use the learned latent space to generate new weights for the same architecture, producing models that are ready for fine-tuning on downstream tasks. We describe each stage in the sections that follow.

3.1. Collecting Remote Sensing Models

Model Retrieval. To train GeoSANE, we need a diverse set of remote sensing models that capture knowledge across multiple sensing modalities, tasks, and architectures. To create such a dataset, we leverage the HuggingFace Hub, which hosts an increasing number of open-source open-weights geospatial models. We query the hub using a broad range of keywords and tags describing both sensing modalities (e.g., Sentinel-1, Sentinel-2, SAR, multispectral) and tasks (e.g., land cover mapping, land cover segmentation, flood detection, disaster response). The complete list of keywords and tags used for model retrieval is provided in the supplementary material. GeoSANE is designed to automatically load and process a wide variety of architectures, including Transformer-based backbones (ViT, Swin, etc.), CNNs (ResNet, UNet, MobileNet, etc.), multimodal radar-optical models, YOLO-style detectors, task-specific models for floods and wildfires, and vision-language models. We exclude corrupted checkpoints, highly undocumented repositories and models requiring unsafe remote code execution. For models with custom or non-standard implementations such as those from TorchGeo [40] or FLAIR [13], we implement custom model loaders, which we will make publicly available together with the final model collection.

Final Model Collection. After filtering, the final collection contains 96 (foundation) remote sensing models, representing approximately 38 billion parameters. Although the number of individual models appears to be smaller than usually found machine learning datasets, the number of model parameters per model is large ranging between hundreds of millions of parameters to billions of parameters.

As a result, even a moderate number of models yields a fair amount of tokens and parameters overall (see Section 5.1), which is sufficient to learn a strong shared representation. As illustrated in Figure 3, our dataset collection cov-

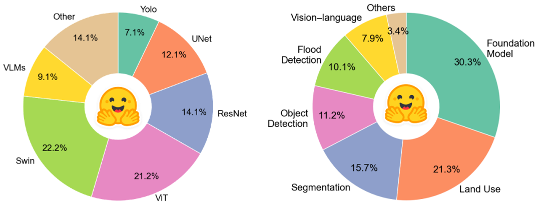

Figure 3. Our model collection retrieved from Hugging Face is diverse as seen in the distribution of model categories in our dataset.

ers a wide range of model categories, including both foundation and task-specific models, providing a comprehensive and representative view of remote sensing models landscape. This model collection serves as the training dataset for GeoSANE, i.e., as input to learn a shared latent representation of neural network weights.

3.2. Learning the Latent Weight Space

While there are various weight space learning methods for model weights generation (see Section 2), GeoSANE follows the encoder-decoder setup of [38] due to its capability to scale to larger model sizes. The core idea of [38] is to tokenize model weights and express an entire model as a sequence of token vectors. Using such a configuration allows the encoder-decoder backbone to learn representations on chunks of the sequences, and therefore enables training on model sequences of different lengths, underlying architectures, and sizes.

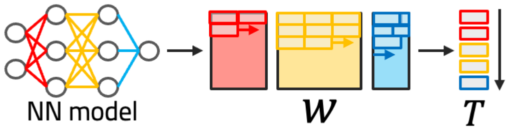

Figure 4. Each weight matrix of a neural network model is tokenized given a fixed-size token. These tokens define the sequence of inputs for autoencoding learning backbone of GeoSANE

Tokenization of Model Weights. To that end, the weights $\mathbf{w}$ of models in the model collection are loaded and reshaped into 2D matrices per layer, then divided into fixed-size tokens $\mathbf{T}_n$ of size d_t (Fig. 4). Zero padding or splitting is applied where needed to ensure uniform dimensions, and a binary mask $\mathbf{M}$ is included to distinguish real parameters from padding. Each token is augmented with a 3D positional embedding $\mathbf{P} = [n, l, k]$ indicating absolute sequence position n, layer index l, and within-layer position k. For training, the tokenized model weights are divided into fixed-length chunks of a specified size (referred to as window) to allow uniform batch sizes and to efficiently process architectures of varying parameter counts. This representation allows processing models of different sizes and architectures in a unified sequence format. For simplicity, we drop the sequence indices n in the following.

Table 1. Comparison between training from scratch and finetuning a GeoSANE-generated model across all benchmark datasets. We report accuracy for single-label classification, mAP for multi-label classification, mIoU for segmentation and mAP@0.5 for object detection. Best results are in **bold**. Δ indicates the absolute improvement over training from scratch.

Dataset	Backbone	from scratch	**GeoSANE**	Δ
EuroSAT [17]	ViT-L	95.0	**99.1**	+4.1
RESISC-45 [5]	ViT-L	78.0	**96.5**	+18.5
fMoW [6]	ViT-L	34.1	**58.9**	+24.8
Sen12Flood [30]	ViT-L	80.4	**85.2**	+4.8
Cal. Wildfires [4]	ViT-L	88.7	**94.9**	+6.2
BigEarthNet [41]	ViT-L	68.3	**88.7**	+20.4
DFC2020 [34]	Swin-B	46.8	**54.3**	+7.5
Spacenet1 [10]	Swin-B	72.2	**78.2**	+6.0
Sen1Floods11 [2]	Swin-B	81.0	**89.6**	+8.6
DIOR [22]	Swin-B	67.5	**79.0**	+11.5

Backbone and Learning Objective. The backbone itself operates as a sequence-to-sequence autoencoder with its bottleneck serving as the shared latent representation of model weights. It consists of an encoder g_θ that maps the input token sequence to a sequence of latent embeddings, $\mathbf{Z} = g_\theta(\mathbf{T}, \mathbf{P})$, and a decoder h_ψ that reconstructs the original tokens from the latent embeddings, $\widehat{\mathbf{T}} = h_\psi(\mathbf{Z}, \mathbf{P})$. To structure the embedding space, a projection head p_ϕ maps the latent embeddings to a lower-dimensional space, $\mathbf{z}_p = p_\phi(\mathbf{Z})$, which is used in a contrastive learning objective. Training is performed on chunks of token sequences with a combination of reconstruction and contrastive loss:

$$\mathcal{L}_{rec} = \|\mathbf{M} \odot (\mathbf{T} - \widehat{\mathbf{T}})\|_2^2, \tag{1}$$

$$\mathcal{L}_c = NTXent(\mathbf{z}_{p,i}, \mathbf{z}_{p,j}), \tag{2}$$

$$\mathcal{L} = (1-\gamma)\mathcal{L}_{rec} + \gamma\mathcal{L}_c. \tag{3}$$

Here, the mask $\mathbf{M}$ is used to separate real parameters with 1 from padding with 0, ensuring that the loss is only computed on actual weights. The contrastive term uses two augmented views i, j of the same model: the first is the original token sequence, while the second is a noised version of it. Both views are processed through the encoder and projection head p_ϕ, and the NT-Xent loss encourages their projected embeddings to be close in latent space while remaining distinct from embeddings of other models. To stabilize training, [38] required normalizing the weights of the models used for training layer-wise during pre-processing which limits the representation learning to models that share the same architecture. This limitation was addressed in recent work [11] proposing to normalize the loss at runtime instead and therefore enabling learning weight-space representations of arbitrary models. Further, it

Table 2. Comparison with existing remote sensing foundation models on multiple remote sensing benchmarks. We report Overall Accuracy for single-label classification, mean Average Precision for multi-label classification, mean intersection over union for segmentation and mean Average Precision (mAP@0.5) for object detection. For GeoSANE, we report the mean ± standard deviation over three independently generated models per prompt (Section 3.3); for all baselines, we report the values reported in the respective papers. Best results are in **bold**, second best are underlined. Δ indicates the absolute improvement over the best baseline

Model	Backbone	Single-label					Multi-label	Segmentation			Object Det.
		EuroSAT	RESISC45	fMoW	Sen12Flood	Wildfires	BigEarthNet	DFC2020	SpaceNet	Sen1Floods11	DIOR
SatMAE [7]	ViT-L	98.9	94.8	58.2	80.3	88.6	86.2	44.1	78.1	–	70.9
Scale-MAE [31]	ViT-L	99.1	95.7	–	82.6	90.8	87.9	–	**78.9**	74.1	73.8
RingMo [42]	Swin-B	–	95.7	–	–	–	–	–	–	–	75.9
CROMA [12]	ViT-L	**99.4**	–	**59.0**	$\underline{83.4}$	$\underline{93.3}$	88.3	$\underline{49.8}$	–	**90.9**	–
GFM [26]	Swin-B	–	–	–	77.9	91.9	86.3	–	–	72.6	–
SkySense [14]	ViT-L	–	–	–	–	–	$\underline{88.6}$	–	–	–	$\underline{78.7}$
DOFA [51]	ViT-L	–	**97.3**	–	–	–	–	–	–	89.4	–
GeoSANE	ViT-L or Swin-B	$\underline{99.1}$±0.2	$\underline{96.5}$ ±0.1	$\underline{58.9}$±0.1	**85.2**±0.3	**94.9**±0.2	**88.7**±0.1	**54.3**±0.3	$\underline{78.2}$±0.3	$\underline{89.6}$±0.1	**79.0**±0.2
Δ		-0.3	-0.8	-0.1	+1.8	+1.6	+0.1	+4.5	-0.7	-1.3	+0.3

demonstrated the feasibility of using models from publicly available model repositories such as Hugging Face as training data, instead of training the weight-space backbone on homogeneous populations of neural network models. This formulation is particularly important in our setting, where the goal is to learn from a broad collection of remote sensing models that vary across architectures, sensing modalities, and tasks.

3.3. Generating new Models from the Latent Space

Once the backbone is trained, GeoSANE can generate weights for new neural network models from the learned representation space. Given a *prompt model* a (e.g., an ImageNet-pretrained ViT-L or Swin-B from the `timm` library), we tokenize and encode its weights $\mathbf{w}_a$ into the latent space to obtain a latent representation $\mathbf{Z}_a = g_\theta(\mathbf{T}_a)$. We then fit a Kernel Density Estimator (KDE) around $\mathbf{Z}_a$ and draw samples $\tilde{\mathbf{z}}$ from this local distribution. This sampling procedure explores nearby regions in the latent space that are structurally similar to the prompt, while being shaped by the geospatial knowledge captured during training. Each sampled latent representation $\tilde{\mathbf{z}}$ is decoded using the decoder to produce synthetic weight tokens $\tilde{\mathbf{T}} = h_\psi(\tilde{\mathbf{z}})$, which are subsequently de-tokenized into neural network weights $\tilde{\mathbf{w}}$. The resulting neural network shares the architecture of the *prompt model* but differs in the actual parameter values, producing a network that is ready for fine-tuning on the target downstream task. Sampling in latent space is inexpensive, as both sampling and decoding require only forward passes. This makes it feasible to generate multiple candidate models and select top-m candidates according to a simple performance criterion before fine-tuning.

In practice, we use ViT-L prompts for classification and Swin-B prompts for segmentation and detection. We generate 10 candidate models per prompt and retain the best m=3 for fine-tuning. In Table 7, we evaluate the generation process using a larger set of prompt models, demonstrating that the method generalizes across backbone families.

4. Downstream Tasks and Datasets

We evaluate models generated by GeoSANE on a diverse set of downstream tasks, covering classification, segmentation, and object detection across multiple sensing modalities (RGB, multispectral, SAR, and SWIR).

Classification We use six diverse datasets: RESISC45 [5] (RGB scene recognition), EuroSAT [17] (multispectral land use classification), fMoW [6] (multispectral scene classification, using 10% of the training set following common practice [7, 25]), BigEarthNet [41] (multispectral multi-label land cover classification), Sen12Flood [30] (SAR-based flood detection), and California Wildfires [3, 4] (SWIR-based wildfire detection). We also evaluate on four classification benchmarks from GEO-Bench [21] (m-EuroSAT, m-BigEarthNet, m-So2Sat, m-Brick-Kiln) following [8], to assess the generalization on standardized remote sensing sets.

Semantic Segmentation We evaluate on three segmentation benchmarks: DFC2020 [34] (multispectral land cover mapping), Sen1Floods11 [2] (SAR-based flood segmentation) and SpaceNet1 [10] (RGB building footprint extraction).

Object Detection For object detection, we use the DIOR dataset [22], which contains 20 object categories in high-resolution RGB aerial imagery.

Together, these datasets cover a wide range of remote sensing tasks, multiple band types, and both single-label and multi-label setups, ensuring that the evaluation is comprehensive. Further dataset details are provided in the supplementary material.

Table 3. Comparison of GeoSANE with pruning and distillation baselines. For *Magnitude Pruning* and *Variational Dropout*, we prune pretrained Remote Sensing Foundation Models (RSFMs) (ScaleMAE [31] and SatMAE [7]) and an ImageNet(IN)-pretrained ViT-L to obtain versions with approximately 11M non-zero parameters (ResNet-18) and 5M (MobileNetV2) non-zero parameters. For *Knowledge Distillation*, the same RSFMs and the ImageNet ViT-L act as teachers, and the student networks are ResNet-18 models with 11M parameters or MobileNetV2 with 3.5M parameters. GeoSANE directly generates models of the target architecture and size. Best results are in **bold**, second best are underlined. Δ indicates the absolute improvement over the best baseline

	Dataset	Magnitude Pruning (MP) Initial Models			Variational Dropout (VP) Initial Models			Knowledge Distillation (KD) Teachers			**GeoSANE**	Δ
		ViT-L (IN)	ScaleMAE	SatMAE	ViT-L (IN)	ScaleMAE	SatMAE	ViT-L (IN)	ScaleMAE	SatMAE		
11M Params	RESISC-45	88.1	87.2	84.9	81.1	81.8	80.6	90.2	90.3	91.1	**92.2**	+1.1
	EuroSAT	97.7	95.4	96.3	95.5	97.0	94.7	97.9	94.0	94.7	**98.7**	+0.8
	fMoW	33.7	36.2	35.4	25.2	32.3	27.1	38.6	43.1	41.9	**53.5**	+10.4
	BigEarthNet	45.6	59.1	62.1	38.2	44.9	49.8	65.7	67.3	66.5	**83.7**	+16.4
	Sen12Flood	77.2	79.0	76.5	75.4	75.3	77.8	79.3	82.1	80.8	**84.0**	+1.9
	Cal. Wildfires	82.9	82.7	84.6	79.1	79.8	81.3	85.2	87.6	88.0	**91.6**	+3.6
3.5M Params*	RESISC-45	64.9	65.6	63.6	61.2	62.0	61.0	67.2	67.5	68.1	**70.0**	+1.9
	EuroSAT	89.2	90.8	91.3	85.5	88.9	90.7	92.1	93.4	94.8	**96.2**	+1.4
	fMoW	15.2	16.7	20.5	16.5	17.8	17.4	23.2	**25.5**	22.1	17.7	-7.8
	BigEarthNet	42.5	41.8	43.9	32.8	34.6	37.1	55.1	60.7	58.5	**73.3**	+12.6
	Sen12Flood	60.1	62.7	62.8	55.9	58.4	59.5	62.1	62.8	64.7	**70.2**	+5.5
	Cal. Wildfires	68.0	69.6	69.9	62.9	63.7	66.2	72.9	74.1	75.4	**75.9**	+0.5

* For MP and VP, models were pruned to approx. 5M parameters instead of 3.5M, as lower sparsity caused model instability.

5. Experiments and Results

5.1. Implementation Details

GeoSANE is implemented as an autoencoder with approximately 900M parameters, where both the encoder and decoder are GPT-2 transformers [29]. Model weights are reshaped and tokenized into fixed-size vectors of dimension 230, resulting in a total of approximately 165M tokens from our remote sensing model collection. To obtain a stronger latent representation, we first pretrain GeoSANE on a larger corpus of general computer vision models from HuggingFace [11] (approximately 700M tokens), and then finetune on the remote sensing tokens. GeoSANE is trained for 150 epochs on a single NVIDIA H100 GPU. We use AdamW [23] with a learning rate of 2×10^{-5}, weight decay of 3×10^{-9}, and a OneCycleLR learning rate schedule. The model is optimized using a combination of reconstruction loss and contrastive guidance, as described in Section 3. The checkpoint with the lowest validation loss is retained for downstream model generation. For completeness, we also report downstream results using models generated from the latent space *before* fine-tuning GeoSANE on remote sensing data (i.e., pretrained only on general computer vision models). These results are provided in the supplementary material.

5.2. Performance of Generated Models

We evaluate GeoSANE to test its ability to generate performant model weights across diverse remote sensing tasks and architectures. Our experiments are designed to answer the following main questions:

Q1. Does GeoSANE provide better initialization than training from scratch ?

Q2. How do models generated by GeoSANE compare to existing remote sensing foundation models?

Q3. Does GeoSANE go beyond model merging by learning relationships in weight space rather than just interpolating weights?

Q4. Can GeoSANE generate strong lightweight models without explicit compression?

Q5. How does GeoSANE improve and generalize across diverse model prompts?

Experimental Setup Unless otherwise specified, classification experiments use GeoSANE-generated ViT-L weights as the base architecture. For segmentation, we attach a lightweight segmentation head to a GeoSANE-generated Swin-B backbone. For object detection, following prior work [14, 42], we use a Faster R-CNN detector [32] with a Swin-B backbone as the feature extractor. For downstream evaluation, all generated models are finetuned for 50 epochs using AdamW as the optimizer. We select the final checkpoint based on the lowest validation loss and report its corresponding test performance.

Q1: Initialization and Fine-tuning Performance

Table 1 compares models generated by GeoSANE with randomly initialized ones of identical architectures and finetuned for the same number of epochs, under similar conditions. Across ten diverse datasets, GeoSANE consistently outperforms training from scratch, with particularly large gains on more challenging, heterogeneous datasets such as fMoW and BigEarthNet, which contain many classes and fine-grained labels. These results show that GeoSANE can serve as an effective initializer for remote sensing models, across various modalities.

Table 4. Comparison of a model generated by GeoSANE against a merged model obtained by combining RSFMs using DARE [53], and against individual RSFMs.

Model	Single-label		Multi-label
	EuroSAT	RESISC45	BigEarthNet
SatMAE [7]	98.9	94.8	86.2
Scale-MAE [31]	**99.1**	95.7	87.9
Merged Model	96.4	86.1	69.0
GeoSANE	**99.1**	**96.5**	**88.7**

Q2: Comparison with Existing RSFMs

Next, we benchmark GeoSANE against many exiting Remote Sensing Foundation Models including SatMAE [7], Scale-MAE [31], CROMA [12], RingMo [42], GFM [26]S, kySense [14] and DOFA [51]. For classification tasks we use ViT-L backbones, and for segmentation and detection tasks we use Swin-B. As shown in Table 2, GeoSANE achieves the best or second-best results across ten benchmarks, matching or surpassing RSFMs. These results show that GeoSANE can reach the same level of performance as models that rely on large-scale pretraining, while generating weights directly from its learned latent representation.

Evaluation on GEO-Bench

We further evaluate GeoSANE on GEO-Bench [21], which provides standardized benchmarks for remote sensing foundation models across multiple modalities and resolutions. As shown in Table 6, GeoSANE achieves the best or second-best performance across all four classification datasets.

Q3: Comparison with Model Merging Methods

Since GeoSANE learns a latent representation from many pretrained models, it is natural to ask whether simpler parameter-space combination techniques could achieve similar performances. A common baseline is model merging, where weights from different networks are combined directly in parameter space. We therefore merge pairs of remote sensing foundation models using the DARE (**D**rop **A**nd **RE**scale) [53] method and compare the resulting merged models to GeoSANE-generated models of the same architecture. As shown in Table 4, GeoSANE achieves consistently higher performance, confirming that learning to generate weights in latent space goes beyond parameter averaging.

Q4: Comparison with Pruning and Distillation for Lightweight Model Generation

A key advantage of GeoSANE is its ability to generate models of different sizes directly in weight space. This is particularly useful in remote sensing, where deployment often requires small and efficient models for on-board processing. To evaluate this, we compare GeoSANE to traditional compression techniques: magnitude pruning [16], variational dropout [27], and knowledge distillation [18]. For pruning and variational dropout, we start from pretrained RSFMs and compress them to ResNet-18 (11M parameters) and MobileNetV2 (3.5M parameters) equivalents, followed by fine-tuning under the same training settings as GeoSANE. For distillation, the same RSFMs serve as teacher models, while ResNet-18 and MobileNetV2 act as students. In contrast, GeoSANE directly generates ResNet-18 and MobileNetV2 weights of the same parameter budgets, requiring neither a large teacher nor iterative pruning. As shown in Table 3, GeoSANE consistently outperforms both pruning and distillation baselines across all datasets and parameter budgets. These results demonstrate that GeoSANE can effectively generate lightweight, high-performing models directly, without relying on compression pipelines or teacher supervision.

Table 5. Comparison between finetuning the prompt models used for generation (ImageNet-pretrained ViT-L for classification and Swin-B for segmentation and detection) and finetuning GeoSANE-generated models of the same architectures.

Dataset	Prompt	**GeoSANE**	Δ
EuroSAT	97.8	**99.1**	+1.3
RESISC45	92.3	**96.5**	+4.2
fMoW	52.4	**58.9**	+6.5
Sen12Flood	84.2	**85.2**	+1.0
Cal. Wildfires	93.5	**94.9**	+1.4
BigEarthNet	82.6	**88.7**	+6.1
DFC2020	47.9	**54.3**	+6.4
Spacenet1	75.8	**78.2**	+2.4
Sen1Floods11	85.2	**89.6**	+4.4
DIOR	73.6	**79.0**	+5.4

Q5: Comparison With Prompt Models and Diversity of Generated Models

GeoSANE-generated models vs. their prompts. We next evaluate whether GeoSANE-generated models improve over the pretrained models used as prompts during generation. For each task, we use an ImageNet-pretrained model as the prompt (ViT-L for classification, Swin-B for segmentation and detection) and generate new weights through GeoSANE. We then fine-tune both the prompt and the generated model under identical settings. As shown in Table 5, GeoSANE consistently yields higher performance, showing that the generation process produces meaningful and effective weight configurations. Since GeoSANE is trained on a diverse collection of remote sensing models, it learns weight patterns that are more aligned with geospatial data, enabling it to produce initializations that outperform

Table 6. Results on four classification tasks from GEO-Bench [21]. All models are trained for 50 epochs. The reported numbers are overall accuracy (OA). The *m-bigearthnet* dataset is evaluated using mAP. For GeoSANE, we report the mean ± standard deviation over three independently generated models per prompt (Section 3.3); for all baselines, we report the values reported in the respective paper [8]. Best results are in **bold**, second best are underlined. Δ indicates the absolute improvement over the best baseline

Method	Backbone	m-eurosat	m-bigearthnet	m-so2sat	m-brick-kiln
SatMAE [7]	ViT-L	96.6	68.3	57.2	98.4
CROMA [12]	ViT-L	96.6	71.9	60.6	98.7
DOFA [51]	ViT-L	96.9	68.0	58.7	98.6
Prithvi-EO 2.0 [43]	ViT-L	96.5	69.0	54.6	98.6
AnySat [1]	ViT-B	95.9	70.3	51.8	98.6
Galileo [45]	ViT-B	97.7	70.7	63.3	98.7
TerraFM [8]	ViT-L	**98.6**	73.1	64.9	**99.0**
GeoSANE	ViT-L	97.7±0.1	**74.2**±0.3	**65.7**±0.2	98.6±0.2
Δ		-0.9	+1.1	+0.8	-0.4

Table 7. We evaluate diverse GeoSANE-generated models on three tasks. All generated models are fine-tuned for 50 epochs.

(a) Classification with GeoSANE-generated Models

Dataset	MobileNet	ResNet-18	ViT-Base	ViT-Large
	3.5M	11M	86M	300M
RESISC-45	70.0	92.2	91.4	96.5
EuroSAT	96.2	98.7	98.4	99.1
fMoW	17.7	53.5	56.7	58.9
BigEarthNet	73.3	83.7	86.9	88.7
Sen12Flood	70.2	84.0	83.1	85.2
Cal. Wildfires	75.9	91.6	93.8	94.9

(b) Semantic Segmentation with GeoSANE-generated Models

Dataset	UNet	Swin-Base
	17M	88M
DFC2020	48.3	54.3
Spacenet1	76.7	78.2
Sen1Floods11	84.2	89.2

(c) Object Detection with GeoSANE-generated Models

Dataset	ResNet-50	Swin-Base	ViT-Large
	26M	88M	300M
DIOR	57.9	79.0	77.4

ImageNet-pretrained prompts.

On-demand Diverse Models Generation Beyond matching the performance of existing RSFMs, GeoSANE also offers a major practical advantage: is not limited to a single backbone family. It can generate weights for a diverse range of architectures and model sizes, including CNNs (ResNet, MobileNet, UNet, YOLO) and Transformers (ViT, Swin). Across classification, segmentation, and detection tasks, these generated models show strong performance, proving that GeoSANE can generalize across very different architectures and tasks. Table 7 summarizes the results for all tasks and backbones. All models are generated by GeoSANE and fine-tuned for 50 epochs.

6. Conclusion

In this work, we introduced GeoSANE, a model foundry that learns geospatial representations directly from the weights of existing models rather than from raw satellite data. By embedding a heterogeneous population of pre-trained remote sensing (foundation) models into a shared latent representation, GeoSANE enables the generation of new model weights on demand, tailored to specific architectures and tasks.

Across classification, segmentation, and detection tasks, GeoSANE consistently improves downstream performance over training from scratch, matches or surpasses state-of-the-art remote sensing foundation models, and is able to generate lightweight networks that outperform pruning- and distillation-based approaches. These results demonstrate that weight-space model generation is a competitive and effective alternative to large-scale pretraining. By learning from models instead of data, GeoSANE offers a scalable path to unify and transfer geospatial knowledge as the diversity and volume of remote sensing models continue to expand.

7. Acknowledgments

let's mention: - SNF remote sensing project (213064) - SPRIND project - who else?

This work was supported in part by ... to D. Borth, and by the US National Science Foundation to S.

Yu under awards NSF 2215542 and NSF 2313151.

References

[1] Guillaume Astruc, Nicolas Gonthier, Clément Mallet, and Loic Landrieu. Anysat: One earth observation model for many resolutions, scales, and modalities. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19530–19540, 2024. 1, 2, 8

[2] Derrick Bonafilia, Beth Tellman, Tyler Anderson, and Erica Issenberg. Sen1floods11: a georeferenced dataset to train and test deep learning flood algorithms for sentinel-1. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 835–845, 2020. 4, 5, 2

[3] California Department of Forestry and Fire Protection. CAL FIRE Incidents. https://www.fire.ca.gov/incidents, . Accessed: 2024-11. 5

[4] California Department of Forestry and Fire Protection. California Fire Perimeters (all). https://catalog.data.gov/dataset/california-fire-perimeters-all-b3436, . Accessed: 2024-11. 4, 5, 2

[5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105:1865–1883, 2017. 4, 5, 2

[6] Gordon A. Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2017. 4, 5, 2

[7] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, M. Burke, D. Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *ArXiv*, abs/2207.08051, 2022. 1, 2, 5, 6, 7, 8

[8] Muhammad Sohail Danish, Muhammad Akhtar Munir, Syed Roshaan Ali Shah, Muhammad Haris Khan, Rao Muhammad Anwer, Jorma Laaksonen, Fahad Shahbaz Khan, and Salman H. Khan. Terrafm: A scalable foundation model for unified multisensor earth observation. *ArXiv*, abs/2506.06281, 2025. 1, 2, 5, 8

[9] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012. 2

[10] Adam Van Etten, David Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series. *ArXiv*, abs/1807.01232, 2018. 4, 5, 2

[11] Damian Falk, Konstantin Schurholt, Konstantinos Tzevelekakis, Léo Meynent, and Damian Borth. Learning model representations using publicly available model hubs. 2025. 2, 4, 6

[12] Anthony Fuller, Koreen Millard, and James R. Green. Croma: Remote sensing representations with contrastive radar-optical masked autoencoders. *ArXiv*, abs/2311.00566, 2023. 1, 2, 5, 7, 8

[13] Anatol Garioud, Nicolas Gonthier, Loic Landrieu, Apolline De Wit, Marion Valette, Marc Poup'ee, Sébastien Giordano, and Boris Wattrelos. Flair: a country-scale land cover semantic segmentation dataset from multi-source optical imagery. *ArXiv*, abs/2310.13336, 2023. 3

[14] Xin Guo, Jiangwei Lao, Bo Dang, Yingying Zhang, Lei Yu, Lixiang Ru, Liheng Zhong, Ziyuan Huang, Kang Wu, Dingxiang Hu, Huimei He, Jian Wang, Jingdong Chen, Ming Yang, Yongjun Zhang, and Yansheng Li. Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27662–27673, 2023. 1, 5, 6, 7

[15] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. *ArXiv*, abs/1609.09106, 2016. 2

[16] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *Neural Information Processing Systems*, 2015. 7

[17] Patrick Helber, Benjamin Bischke, Andreas R. Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12:2217–2226, 2017. 4, 5, 2

[18] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 7

[19] Johannes Jakubik, Sujit Roy, C. E. Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, Daiki Kimura, Naomi Simumba, Linsong Chu, S. Karthik Mukkavilli, Devyani Lambhate, Kamal Das, Ranjini Bangalore, Dario Oliveira, Michal Muszynski, Kumar Ankur, Muthukumaran Ramasubramanian, Iksha Gurung, Sam Khallaghi, Hanxi (Steve) Li, Michael Cecil, Maryam Ahmadi, Fatemeh Kordi, Hamed Alemohammad, Manil Maskey, Raghu Ganti, Kommy Weldemariam, and Rahul Ramachandran. Foundation Models for Generalist Geospatial Artificial Intelligence. *Preprint Available on arxiv:2310.18660*, 2023. 1, 2

[20] Johannes Jakubik, Felix Yang, Benedikt Blumenstiel, Erik Scheurer, Rocco Sedona, Stefano Maurogiovanni, Jente Bosmans, Nikolaos Dionelis, Valerio Marsocci, Niklas Kopp, Rahul Ramachandran, Paolo Fraccaro, Thomas Brunschwiler, Gabriele Cavallaro, Juan Bernabé-Moreno, and Nicolas Long'ep'e. Terramind: Large-scale generative multimodality for earth observation. *ArXiv*, abs/2504.11171, 2025. 1, 2

[21] Alexandre Lacoste, Nils Lehmann, Pau Rodríguez López, Evan D. Sherwin, Hannah Rae Kerner, Bjorn Lutjens, Jeremy A. Irvin, David Dao, Hamed Alemohammad, Alexandre Drouin, Mehmet Gunturkun, Gabriel Huang, David Vázquez, Dava Newman, Yoshua Bengio, Stefano Ermon, and Xiao Xiang Zhu. Geo-bench: Toward foundation models for earth monitoring. *ArXiv*, abs/2306.03831, 2023. 5, 7, 8

[22] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ArXiv*, abs/1909.00133, 2019. 4, 5, 2

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 6

[24] Siqi Lu, Junlin Guo, James Zimmer-Dauphinee, Jordan M. Nieusma, Xiao Wang, Parker VanValkenburgh, Steven A. Wernke, and Yuankai Huo. Vision foundation models in remote sensing: A survey. 2024. 1

[25] Oscar Mañas, Alexandre Lacoste, Xavier Giró i Nieto, David Vázquez, and Pau Rodríguez López. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9394–9403, 2021. 2, 5

[26] Matias Mendieta, Boran Han, Xingjian Shi, Yi Zhu, Chen Chen, and Mu Li. Towards geospatial foundation models via continual pretraining. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16760–16770, 2023. 5, 7

[27] Dmitry Molchanov, Arsenii Ashukha, and Dmitry P. Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, 2017. 7

[28] William S. Peebles, Ilija Radosavovic, Tim Brooks, Alexei A. Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *ArXiv*, abs/2209.12892, 2022. 1, 2

[29] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 6

[30] C Rambour, N Audebert, E Koeniguer, B Le Saux, M Crucianu, and M Datcu. Sen12-flood: a SAR and Multispectral Dataset for Flood Detection. *IEEE: Piscataway, NJ, USA*, 2020. 4, 5, 2

[31] Colorado Reed, Ritwik Gupta, Shufan Li, Sara Brockman, Christopher Funk, Brian Clipp, Salvatore Candido, Matthew Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4065–4076, 2022. 1, 2, 5, 6, 7

[32] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 6

[33] Linus Scheibenreif, Joëlle Hanna, Michael Mommert, and Damian Borth. Self-supervised Vision Transformers for Land-cover Segmentation and Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1422–1431, 2022. 2

[34] Michael Schmitt, Lloyd Hughes, Pedram Ghamisi, Naoto Yokoya, and Ronny Hänsch. 2020 ieee grss data fusion contest, 2019. 4, 5, 2

[35] Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Hyper-representations: Self-supervised representation learning on neural network weights for model characteristic prediction. 2021. 1, 2

[36] Konstantin Schürholt, Boris Knyazev, Xavier Gir'o i Nieto, and Damian Borth. Hyper-representations as generative models: Sampling unseen neural network weights. *ArXiv*, abs/2209.14733, 2022. 1, 2

[37] Konstantin Schürholt, Giorgos Bouritsas, Eliahu Horwitz, Derek Lim, Yoav Gelberg, Bo Zhao, Allan Zhou, Damian Borth, and Stefanie Jegelka. Neural network weights as a new data modality. In *ICLR 2025 Workshop Proposals*, 2024. 1

[38] Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning. *ArXiv*, abs/2406.09997, 2024. 1, 2, 4

[39] Bedionita Soro, Bruno Andreis, Hayeon Lee, Song Chong, Frank Hutter, and Sung Ju Hwang. Diffusion-based neural network weights generation. *ArXiv*, abs/2402.18153, 2024. 1, 2

[40] Adam J. Stewart, Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, and Arindam Banerjee. Torchgeo: deep learning with geospatial data. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2021. 3

[41] Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5901–5904, 2019. 4, 5, 2

[42] Xian Sun, Peijin Wang, Wanxuan Lu, Zicong Zhu, Xiaonan Lu, Qi He, Junxi Li, Xuee Rong, Zhujun Yang, Hao Chang, Qinglin He, Guang Yang, Ruiping Wang, Jiwen Lu, and Kun Fu. Ringmo: A remote sensing foundation model with masked image modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–22, 2023. 2, 5, 6, 7

[43] Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, orsteinn Elí Gíslason, Benedikt Blumenstiel, Rinki Ghosal, Pedro Henrique de Oliveira, João Lucas de Sousa Almeida, Rocco Sedona, Yanghui Kang, Srija Chakraborty, Sizhe Wang, Ankur Kumar, Myscon Truong, Denys Godwin, Hyunho Lee, Chia-Yu Hsu, Ata Akbari Asanjan, Besart Mujeci, Trevor Keenan, Paulo Arévalo, Wenwen Li, Hamed Alemohammad, Pontus Olofsson, Christopher Hain, Robert Kennedy, Bianca Zadrozny, Gabriele Cavallaro, Campbell Watson, Manil Maskey, Rahul Ramachandran, and Juan Bernabé-Moreno. Prithvi-eo-2.0: A versatile multi-temporal foundation model for earth observation applications. *ArXiv*, abs/2412.02732, 2024. 1, 2, 8

[44] Ramon Torres, Paul Snoeij, Dirk Geudtner, David Bibby, Malcolm Davidson, Evert Attema, Pierre Potin, BjÖrn Rommen, Nicolas Floury, Mike Brown, et al. Gmes sentinel-1 mission. *Remote sensing of environment*, 120:9–24, 2012. 2

[45] Gabriel Tseng, Anthony Fuller, Marlena Reil, Henry Herzog, Patrick Beukema, Favyen Bastani, James R. Green, Evan Shelhamer, Hannah Kerner, and David Rolnick. Galileo: Learning global&local features of many remote sensing modalities. 2025. 8

[46] Kaitian Wang, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion. *ArXiv*, abs/2402.13144, 2024. 1, 2

[47] Yi Wang, Nassim Ait Ali Braham, Zhitong Xiong, Chenying Liu, Conrad M. Albrecht, and Xiao Xiang Zhu. Ssl4eo-s12: A large-scale multi-modal, multi-temporal dataset for self-supervised learning in earth observation. *ArXiv*, abs/2211.07044, 2022. 2

[48] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7949–7961, 2021. 3

[49] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *ArXiv*, abs/2203.05482, 2022. 3

[50] Aoran Xiao, Weihao Xuan, Junjue Wang, Jiaxing Huang, Dacheng Tao, Shijian Lu, and Naoto Yokoya. Foundation models for remote sensing and earth observation: A survey. *ArXiv*, abs/2410.16602, 2024. 1

[51] Zhitong Xiong, Yi Wang, Fahong Zhang, Adam J. Stewart, Joelle Hanna, Damian Borth, Ioannis Papoutsis, Bertrand Le Saux, Gustau Camps-Valls, and Xiao Xiang Zhu. Neural plasticity-inspired multimodal foundation model for earth observation. 2024. 1, 2, 5, 7, 8

[52] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Neural Information Processing Systems*, 2023. 3

[53] Le Yu, Yu Bowen, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ArXiv*, abs/2311.03099, 2023. 3, 7

GeoSANE: Learning Geospatial Representations from Models, Not Data

Supplementary Material

8. Effect of Remote Sensing Fine-tuning on GeoSANE Model Generation.

To assess the contribution of remote sensing–specific finetuning in GeoSANE's latent weight space, we compare models generated from two variants of the backbone: (1) `CV-only`, where GeoSANE is trained only on general computer vision models, and (2) `CV+RS`, where GeoSANE is additionally finetuned on our remote sensing (RS) model collection. For each setting, we generate weights for the same target architectures (ViT-L) and finetune them identically on downstream tasks. This allows us to isolate the effect of training GeoSANE on remote sensing weights on the quality of the generated models. Results are shown in Table 8.

When comparing Table 8 with the results in Table 5 (models prompt vs. GeoSANE-generated models), we observe that models generated without remote sensing finetuning (`CV-only`) perform similarly to their ImageNet-pretrained model prompts. Since these model prompts are themselves ImageNet models, the latent space trained only on computer vision models provides limited additional benefit. In contrast, with additional training on remote sensing model weights `CV+RS`, GeoSANE-generated models clearly outperform their model prompts. This demonstrates that training on remote sensing model weights is essential: it injects geospatial structure into the latent space and enables the generation of initializations that outperform ImageNet-pretrained prompts.

Table 8. Downstream performance of models generated by GeoSANE trained without (`CV-only`) and with (`CV+RS`) remote sensing model weights.

Dataset	`CV-only`	`CV+RS`	Δ
EuroSAT	97.8	**99.1**	+1.3
RESISC45	93.4	**96.5**	+3.1
fMoW	53.1	**58.9**	+5.8
Sen12Flood	82.9	**85.2**	+2.3
Cal. Wildfires	92.2	**94.9**	+2.7
BigEarthNet	83.0	**88.7**	+5.7
DFC2020	48.6	**54.3**	+5.7
Spacenet1	75.3	**78.2**	+2.9
Sen1Floods11	86.0	**89.6**	+3.6
DIOR	73.7	**79.0**	+5.3

9. Downstream Datasets Sizes

Table 9 provides an overview of the evaluation datasets used in this work, covering scene classification, segmentation, and object detection tasks. For each dataset, we report the number of classes, label type, input channels, and number of samples. We also include the spatial resolution at which we train our models; when datasets provide images at different native resolutions, we uniformly resize them to the sizes listed in the table to ensure consistent training.

10. Additional Details on Model Collection Retrieval and Automatic Loader

Listing 1. Tags and keywords used for models retrieval.

```
[remote sensing, remote-sensing, NDVI, DSM,
remotesensing, earth observation, enMAP,
earth-observation, EO, satellite, LiDAR,
satellites, satellite imagery, S1, S2,
satellite-imagery, aerial, aerial imagery,
aerial-imagery, Sentinel-1, Sentinel 1,
Sentinel-2, Sentinel 2, SAR, landsat, MODIS,
synthetic-aperture-radar, multispectral,
multi-spectral, hyperspectral, cloud-mask,
hyper-spectral, enmap, vegetation-index,
AVIRIS, VIIRS, Pleiades, PlanetScope,
WorldView, drone, UAV, CubeSat, rsfm, RSFM,
satMAE, scalemae, satmae, dofa, optical,
radar, landcover, land-cover, land cover,
land use, land-use, urban mapping,
urban land cover, deforestation, snow cover,
flood detection, flood mapping, wildfire,
fire detection, glacier monitoring, DFC2023,
hazard mapping, coastal erosion, crop
monitoring, precision agriculture, air
quality, natural disasters, marine debris,
disaster response, soil sealing, methane
detection, building extraction, road
extraction, ai4eo, ml4eo, torchgeo, eo-learn,
rasterio, geopandas, earthengine, EuroSAT,
BigEarthNet, xView, FloodNet, SpaceNet,
so2sat, bigearthnet, brick-kiln, forestnet,
pv4ger, RESISC-45, pv4ger-seg, chesapeake,
cashew-plant, Crop Types, NeonTree, Cattles,
SegMunich, UC Merced, AID, FAIR1M, DIOR,
iSAID, ISPRS Potsdam, LEVIR-CD, BurnScars,
MADOS, PASTIS, Sen1Floods11, DynamicEarthNet,
FiveBillionPixels, CTM-SS, SpaceNet7, NAIP,
AI4Farms, METER-ML, fMoW, MLRSNet, WHU-RS19,
Optimal-31, AiRound, CV-BrCT, SpaceNet2,
INRIA Aerial, GID-15, DFC2020, Dynamic World,
MARIDA, WHU Aerial, Vaihingen, OSCD, DSIFN]
```

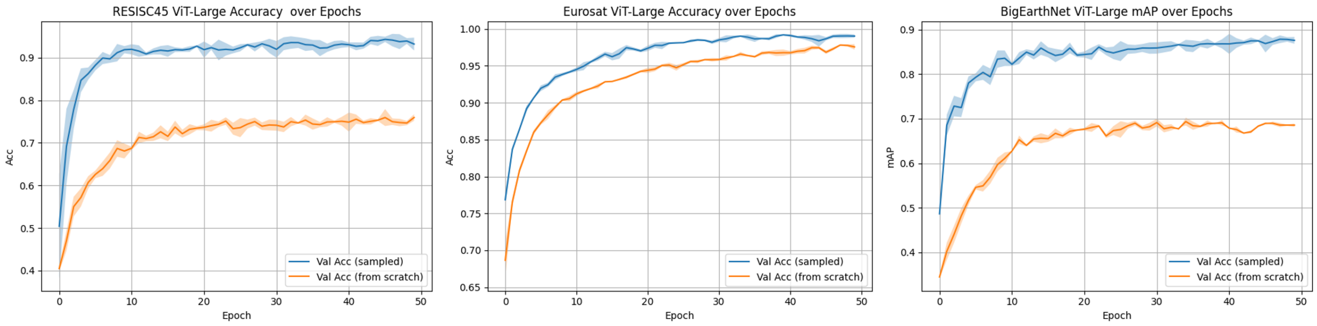

Figure 5. Convergence comparison between GeoSANE-initialized models and models trained from scratch.

Task	Dataset	# of classes	Labels (per image)	Size	Channels	# of samples
Classification	RESISC45 [5]	45	single	256 × 256	RGB	31.5K
	EuroSAT [17]	10	single	64 × 64	Multispectral	27K
	fMoW [6]	63	single	512 × 512	Multispectral	$> 1M$
	BigEarthNet [41]	19	multiple	120 × 120	Multispectral	590K
	Sen12Flood [30]	2 (binary)	single	256 × 256	SAR	16K
	California Wildfires [4]	2 (binary)	single	224 × 224	SWIR	20K
Segmentation	DFC2020 [34]	8	multiple	256 × 256	Multispectral	5K
	Sen1Floods11 [2]	2 (binary)	single	512 × 512	SAR	5K
	Spacenet1 [10]	2 (binary)	single	400 × 432	RGB	7K
Obj. Detection	DIOR [22]	20	multiple	800 × 800	RGB	23.5K

Table 9. Overview of datasets, tasks, label types, and channels.

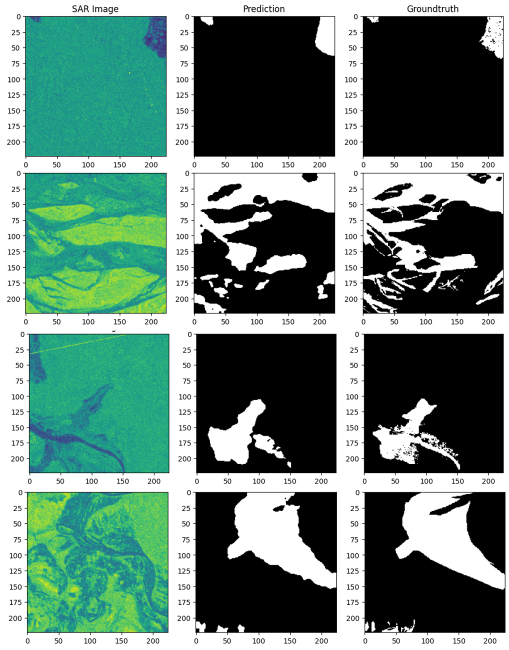

Figure 6. Qualitative Results of the Flood Segmentation task on the Sen1Floods11 dataset

As outlined in the main paper, we build the GeoSANE model collection by querying the HuggingFace Hub using a comprehensive set of modality-, task-, and dataset-specific keywords (see Listing 1). Here we provide the additional technical details required for full reproducibility. Because many remote sensing repositories on HuggingFace lack consistent metadata, we extend the search beyond tags by matching the same keyword vocabulary against repository names, model filenames, and model-card text. This procedure allows us to recover models even when authors provide incomplete or missing tags. The full list of retrieval keywords used in our search is included above.

A second practical challenge is that a substantial fraction of the collected models (particularly those originating from TorchGeo, SSL4EO, or various institutional releases) do not provide configuration files or explicit architectural specifications. To ensure that all such models can be loaded in a unified manner, we implement an automatic architecture reconstruction mechanism. Instead of relying on external configs, the loader infers the backbone type and input dimensionality directly from the structure of the `state_dict`. Convolutional backbones (e.g., ResNet-type models) are identified through their stem convolutions, whose tensor shapes reveal both the expected number of input channels and the architectural family. Transformer-based models (e.g., ViT, Swin) are detected through their patch-embedding layers, from which

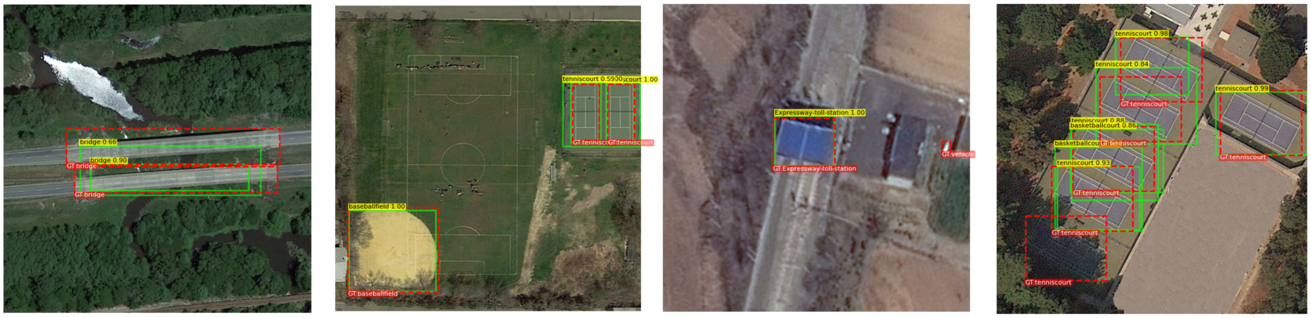

Figure 7. Qualitative Results of the Object Detection task on DIOR dataset

the loader extracts both in_chans and the embedding dimension. When the checkpoint structure remains ambiguous, we fall back on controlled filename heuristics (e.g., detecting multispectral, SAR, RGB, or mission-specific Sentinel/Landsat naming patterns) to infer sensing modality and band count. Using these inferred attributes, the loader reconstructs the closest matching architecture from timm or segmentation_models_pytorch, and loads the checkpoint. This procedure enables GeoSANE to handle heterogeneous, partially documented checkpoints in a consistent and fully automated way. All loader code will be released alongside the final model collection.

11. Qualitative Results

Aside from quantitative results, we also provide qualitative examples. Following the same procedure described in the experimental setup, we first generate a Swin-B backbone with GeoSANE, then attach the appropriate task-specific head (a Faster R-CNN detection head for DIOR, or a segmentation head for Sen1Floods11). Figure 7 shows some object detection outputs on the DIOR dataset, while Figure 6 displays semantic flood segmentation results on Sen1Floods11.

12. UMAP Visualization of the Latent Weight Space

To better understand the structures learned by GeoSANE, we visualize (Figure 8) the latent representations of all models in our remote sensing collection using UMAP. For each model, we extract its full latent embedding sequence, sample 100 tokens randomly, and project these vectors to 2D using UMAP. When colored by architecture (top), the embedding shows clear structural organization: models sharing similar backbone types, such as ViTs, UNets, Swins or ResNets form well-separated clusters. This indicates that GeoSANE's latent space preserves meaningful distinctions between architectural families. When colored by sensing modality (bottom), the same embedding shows modality-related structure: models having similar inputs (e.g., SAR or Landsat) tend to appear in nearby regions of the space. Together, these observations show that GeoSANE organizes heterogeneous remote sensing models into meaningful latent groups.

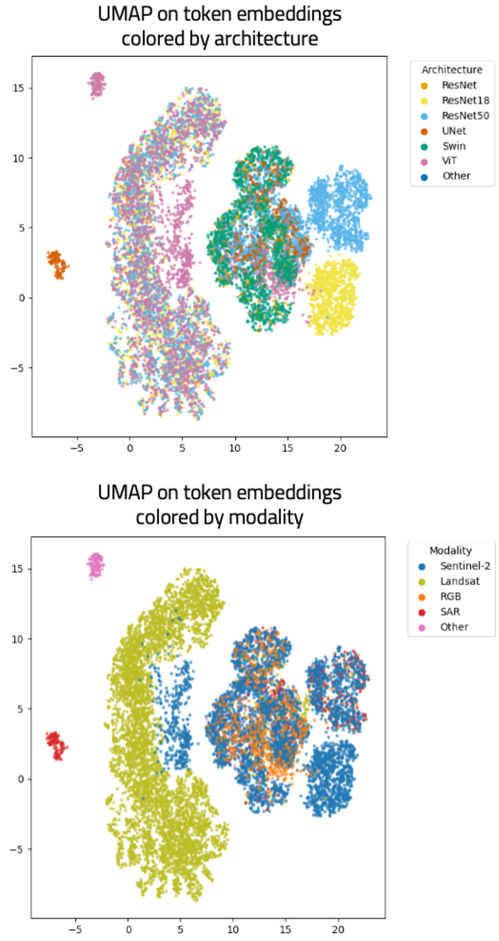

Figure 8. UMAPs Visualization of the latent weight space of GeoSANE, colored by architecture (top) and modality (bottom).

13. Convergence Behavior of GeoSANE-generated Models

We study the convergence behavior of models initialized with GeoSANE compared to models trained from scratch. Figure 5 reports the validation performance over epochs for three representative datasets (RESISC45, EuroSAT, BigEarthNet), all using a ViT-L backbone. In every case, GeoSANE-generated initializations achieve strong accuracy within the first few epochs and maintain a consistent lead throughout training. This shows that GeoSANE does not only produce models competitive with remote sensing foundation models, but it also consistently accelerates optimization; which is an important advantage in settings with limited compute or training budgets.

14. Model Checkpoints Used for Training

Table 12 provides the complete list of model checkpoints used to train GeoSANE. For each model, we report the primary task 11 and the sensing modalities 10 used during training when this information is available from the original repository or documentation.

Modality	Number of Models
Multispectral	41
RGB	14
Multimodal	9
SAR	6
DEM	2
Unknown	24

Table 10. Distribution of modalities for the models used to train GeoSANE.

Task	Number of Models
Self-supervised representation learning	36
Semantic segmentation	15
Object detection	7
Classification	4
Generative models	2
Regression	1
Unknown	31

Table 11. Distribution of primary tasks for the models used to train GeoSANE.

Checkpoint	Checkpoint	Checkpoint
jaychempan__EarthSynth	azdin__llava-onevision-weather-dora	mayrajeo__marine-vessel-yolo
torchgeo__presto	azdin__llava-onevision-weather-qlora	torchgeo__core-dino
DevPanda004__PrithviFlood	azdin__qwen2-vl-weather-adalora	torchgeo__delineate-anything
Mahadih534__YoloV8-VisDrone	azdin__qwen2-vl-weather-dora	torchgeo__delineate-anything-s
Mahadih534__yolov8_ship_det_satellite	banghyunmin__Thermal_Video_Detection	torchgeo__yolo11s_marine_vessel_detection
RedbeardNZ__LatentSync-1.6	banghyunmin__thermal-people-yolov11n	wangyi111__Copernicus-FM
azdin__llava-onevision-weather-adalora	mayrajeo__marine-vessel-detection-yolov8	IGNF__FLAIR-HUB_LC-A_IR_convnextv2base-unet
IGNF__FLAIR-HUB_LC-A_IR_convnextv2base-upernet	IGNF__FLAIR-HUB_LC-A_IR_convnextv2tiny-upernet	IGNF__FLAIR-HUB_LC-A_IR_swinbase-unet
IGNF__FLAIR-HUB_LC-A_IR_swinbase-upernet	IGNF__FLAIR-HUB_LC-A_IR_swinlarge-upernet	IGNF__FLAIR-HUB_LC-A_IR_swinsmall-upernet
IGNF__FLAIR-HUB_LC-A_IR_swintiny-upernet	IGNF__FLAIR-HUB_LC-A_RGB_swinbase-upernet	IGNF__FLAIR-HUB_LC-A_RGB_swinlarge-upernet
IGNF__FLAIR-HUB_LC-A_RGB_swinsmall-upernet	IGNF__FLAIR-HUB_LC-A_RGB_swintiny-upernet	IGNF__FLAIR-HUB_LC-D_swinbase-upernet
IGNF__FLAIR-HUB_LC-F_swinbase-upernet	IGNF__FLAIR-HUB_LC-I_swinbase-upernet	IGNF__FLAIR-HUB_LC-L_swinbase-upernet
IGNF__FLAIR-HUB_LPIS-A_swinbase-upernet	IGNF__FLAIR-HUB_LPIS-I_swinbase-upernet	IGNF__FLAIR-HUB_LPIS-J_swinbase-upernet
Jabasingh__VCTI	torchgeo__core-dino	torchgeo__satlas
chrimerss__flood-foundation-prithvi-100m	torchgeo__croma	torchgeo__seco-eco
chrimerss__flood-foundation-prithvi-300m	torchgeo__decur	torchgeo__seco-eco-ndvi
chrimerss__flood-foundation-prithvi-600m	torchgeo__delineate-anything	torchgeo__sentinel1_unet_effb4_openearthmap_sar
chrimerss__flood-foundation-prithvi-tiny	torchgeo__delineate-anything-s	torchgeo__ssl4eo_landsat
chrimerss__flood-foundation-resnet101-unet	torchgeo__dofa	torchgeo__swin_v2_b_naip_rgb_satlas
chrimerss__flood-foundation-resnet152-unet	torchgeo__earthloc	torchgeo__swin_v2_b_sentinel2_rgb_satlas
chrimerss__flood-foundation-resnet50-unet	torchgeo__fields-of-the-world	torchgeo__unet_resnet34_oam_rgb_tcd
galeio-research__OceanSAR-1	torchgeo__ftw	torchgeo__unet_resnet50_oam_rgb_tcd
galeio-research__OceanSAR-1-tengeop	torchgeo__resnet18_sentinel2_all_moco	torchgeo__vit_base_patch32_224_skyclip_50pct
galeio-research__OceanSAR-1-wave	torchgeo__resnet18_sentinel2_rgb_moco	torchgeo__vit_large_patch14_224_clip_laionrs
galeio-research__OceanSAR-1-wind	torchgeo__resnet18_sentinel2_rgb_seco	torchgeo__vit_large_patch14_224_skyclip_30pct
mrm8488__convnext-tiny-finetuned-eurosat	torchgeo__resnet50_fmow_rgb_gassl	torchgeo__vit_large_patch14_224_skyclip_50pct
openclimatefix__power_perceiver	torchgeo__resnet50_landsat7_l2_all_moco	torchgeo__vit_large_patch16_224_fmow_rgb_scalemae
pszemraj__convnextv2-nano-22k-384-boulderspot	torchgeo__resnet50_sentinel1_all_moco	torchgeo__vit_small_patch16_224_sentinel2_all_dino
quantum-leap-vcti__VCTI-RoBERTa-Fiber	torchgeo__resnet50_sentinel2_all_dino	torchgeo__vit_small_patch16_224_sentinel2_all_moco
swardiantara__drone-term-extractor	torchgeo__resnet50_sentinel2_all_moco	torchgeo__yolo11s_marine_vessel_detection
torchgeo__ai4g_flood	torchgeo__resnet50_sentinel2_rgb_moco	Burdenthrive__cloud-detection-segformer-mit_b4-RGB
torchgeo__copernicus-fm	torchgeo__resnet50_sentinel2_rgb_seco	Burdenthrive__cloud-detection-unet-regnetzd8

Table 12. Remote Sensing Model checkpoints used to train GeoSANE.