ATOMIZER-IO: BEYOND PIXELS, PATCHES AND GRIDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most vision architectures assume that observations lie on a regular grid, an effective abstraction for natural images but a restrictive one for sensing data whose channels, temporal sampling, spatial resolution, and geometry can vary. Generic set-based architectures remove the grid, but also remove useful spatial inductive biases. We introduce Atomizer-IO, an architecture that places observations first and derives structure from their physical relationships. Building on top of an atomic representation of the data, each observation is described by its measurement and acquisition metadata, while local cross-attention maps observations to anchor points that can be arbitrarily placed. We evaluate this design by progressively relaxing the grid assumption, from varying input raster configurations and incomplete channel sets to flexible output density and, ultimately, inputs without a raster grid. Atomizer-IO is competitive with flexible EO-specific architectures on most tasks, while offering post-training control over inference cost and competitive compute–performance trade-offs. The same formulation extends without architectural redesign to unordered 3D point clouds, showing that the atomic interface generalizes beyond regular raster inputs. These results suggest that pixels, patches, and grids do not need to define the interface of a sensing architecture.

1 INTRODUCTION

Most computer vision models, including convolutional neural networks (CNN) (LeCun et al., 1989) and vision transformers (ViT) (Dosovitskiy et al., 2020), see the world as a regular grid. They operate on tensors whose channels and spatial organization are known in advance. Natural images fit this interface well: RGB channels have fixed semantics, pixels lie on a regular lattice, and neighboring entries have a constant spatial offset. Convolutions exploit this structure explicitly, while ViTs retain the same regular input organization even as they relax locality. In many sensing domains, however, spectral content, spatial sampling, acquisition time, and even geometry vary, making a fixed tensor overly restrictive.

Earth observation (EO) is an acute instance of this mismatch. Data in this field is heterogeneous by nature, since different sensors observe different physical quantities, at different resolutions, and at different times. Channel position therefore provides no stable physical identity across sensors. A short-wave infrared measurement sensitive to moisture and a SAR backscatter measurement sensitive to surface roughness may occupy analogous tensor positions while representing fundamentally different physical quantities. In standard architectures, measurement identity is therefore encoded implicitly through channel position rather than explicitly through physical characteristics (Jakubik et al., 2025). Measurement identity is only one axis: resolution, acquisition time, channel availability, and geometry vary too.

Recent EO architectures increasingly accommodate these sources of variation through dedicated mechanisms. Rather than changing the underlying representation, they typically adapt a tensor- or patch-based interface to each source of variation. Within this fixed interface, a missing channel must be padded (Sumbul et al., 2025) with values the model cannot distinguish from real measurements, differing resolutions resampled or handled by resolution-aware processing (Marsocci et al., 2024), and irregular observations forced onto a grid before entering the model (Weinstein et al., 2021). Each mechanism compensates for an assumption introduced by the representation itself.

Our proposed Atomizer-IO reverses this order: *observations come first, structure second*. Each observation is represented independently by its measurement and acquisition metadata, without re-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

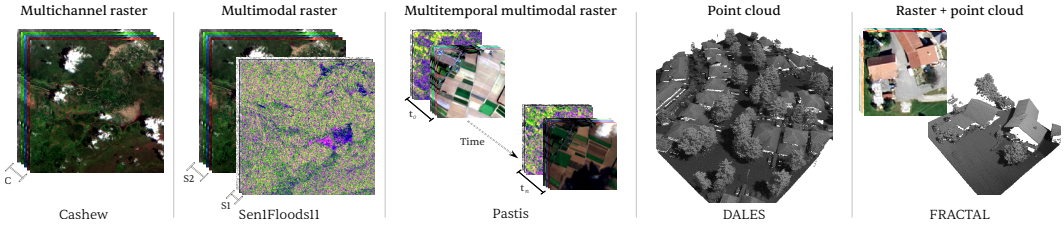


Figure 1: Sensing configurations considered in this work. All are processed through the same atomic interface, without rasterization, resampling, or modality-specific branches.

quiring a fixed channel count, tensor shape, or neighboring measurement. Structure is introduced afterwards from physical relationships between observations: locality and relative geometry are defined directly in physical coordinates rather than inherited from tensor position.

Atomizer-IO implements this principle through an atomic encoder–processor–decoder architecture. Following Atomizer (de Turckheim et al., 2025), each observation is represented as an *atom*; for raster data, one atom corresponds to one measurement from one spectral band, at one pixel location and one acquisition time. The encoder aggregates atoms into spatially anchored latent vectors, the processor lets these latents interact, and a query-based decoder maps local latent neighborhoods to requested output locations. Because atomic granularity can produce millions of observations per scene, each latent attends only to a local neighborhood in physical space, while the number of spatial latents scales with the number of input observations. The same local aggregation serves as both input and output interface, allowing channel sets, input resolution, and output density to vary while extending the same architecture from regular rasters to unordered 3D point clouds.

We evaluate this design by progressively relaxing the assumptions of a fixed grid. We first assess competitiveness on conventional raster EO tasks, then vary available channels and output-query density, and finally remove the raster assumption with unordered LiDAR and joint image–point observations.

Our main contributions are:

- **A physically grounded atomic representation** in which measurement identity is described explicitly rather than tied to tensor position. A single trained model spans sensor configurations that a fixed-channel architecture can only reach by padding or redesign.
- **A scalable, structured set-to-latent architecture** that uses the same local geometric operation to encode observations and decode arbitrary query locations, removing the need for a fixed grid on either side.
- **A single architecture across raster and irregular sensing geometries**, spanning classification, regression, 2D and 3D segmentation, and optical, radar, and LiDAR observations. All models are trained from scratch under matched conditions, isolating architectural differences from pretraining.

2 Related Work

Regular grids supply useful spatial structure, including locality, stable neighborhoods, and regular geometric relationships, that grid-based architectures can exploit directly. In Earth observation, the representation must additionally accommodate several physical axes: spatial resolution determines how space is sampled, spectral bands sample the electromagnetic spectrum, and acquisition times sample an evolving scene.

Recent EO models increasingly expose these physical properties to grid-based architectures through dedicated mechanisms. DOFA (Xiong et al., 2024), SenPa-MAE (Prexl & Schmitt, 2024), and Panopticon (Waldmann et al., 2025) primarily address spectral variation, while FlexiMo (Li et al., 2025) additionally handles spatial variation but does not natively model temporal sequences. Scale-MAE (Reed et al., 2023) incorporates ground sampling distance, AnySat (Astruc et al., 2024) defines patches in physical rather than pixel units, and Galileo (Tseng et al., 2025) and Prithvi (Szwarcman et al., 2025) introduce temporal structure. RAMEN (Houdré et al., 2026) and UniverSat (Perron et al., 2026) jointly accommodate spectral, spatial, and temporal variation within a single architecture, making them the closest flexible EO baselines to our framework.

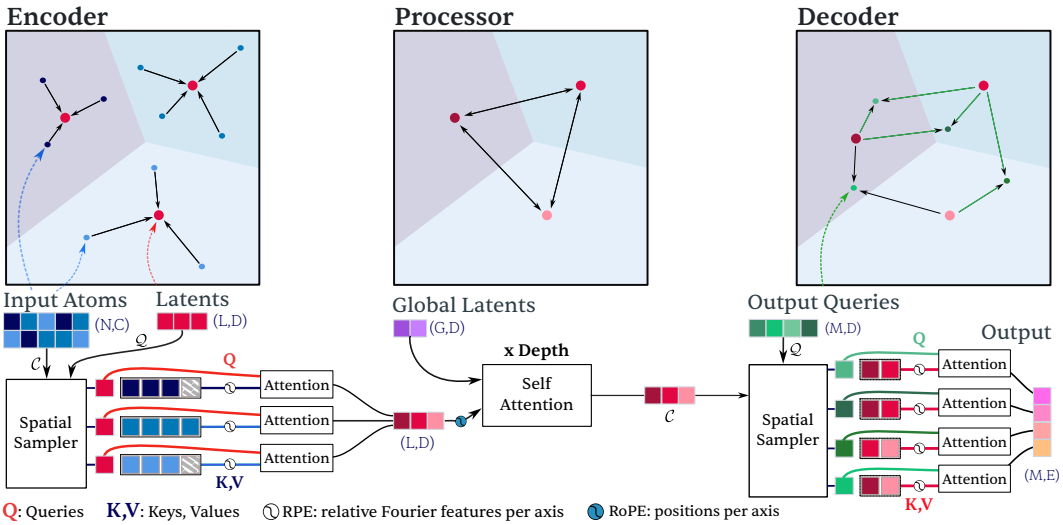


Figure 2: The Atomizer-IO architecture. Atoms are locally aggregated into a compact spatial representation, processed globally in latent space, and decoded through spatial output queries. The same local context-to-query operation is used for encoding and decoding, with spatial relationships defined in physical coordinates rather than by a fixed input grid.

A separate line of work relaxes this requirement by changing the unit on which the model operates. Presto (Tseng et al., 2023) discards image structure entirely, modeling individual pixel time series and relying on temporal and multimodal signal instead, while Atomizer (de Turckheim et al., 2025) retains atomic measurements as the input representation and processes them as an unordered set. More generally, Set Transformer (Lee et al., 2019) showed that attention can operate directly on unordered sets, while Perceiver (Jaegle et al., 2021b) introduced cross-attention to a compact latent bottleneck and Perceiver IO (Jaegle et al., 2021a) extended this paradigm to query-defined outputs.

Moving beyond grids does not mean abandoning spatial structure. 3D point-cloud methods routinely operate on unordered, sparsely distributed points that do not fit the dense tensor representations common in 2D vision (Guo et al., 2019). Various alternatives have been proposed: PointNet (Qi et al., 2017a) operates on sets, PointNet++ (Qi et al., 2017b) and KPConv (Thomas et al., 2019) on multi-scale Euclidean neighborhoods, and Superpoint Transformer (Robert et al., 2023), not unlike our work, on geometric graphs over 3D partitions. However, these approaches are not designed to jointly accommodate multi-spectral, multi-temporal, and multi-resolution data. This creates a tension: grid-based models preserve useful spatial structure but require observations to conform to a fixed layout, while set-based models gain flexibility but provide no geometry by default. Atomizer-IO resolves this by enabling reasoning over an arbitrary Voronoi tessellation of space. Sensor-specific gridded structures are confined to the interfaces, where input observations are either read in, or output predictions are queried out.

3 Methodology

Atomizer-IO builds on Atomizer (de Turckheim et al., 2025), which decomposes an input into an unordered set of atoms. Where Atomizer lets non-spatial latents cross-attend to all atoms, and only for image classification, Atomizer-IO introduces locality. Each latent corresponds to a spatial anchor and attends only to nearby atoms, while still exchanging information with every other latent. Spatially dense outputs are produced by querying the latents near any requested location.

This section follows one observation through the model (Figure 2): token construction (Section 3.1), neighborhood assignment (Section 3.2), aggregation into a spatial latent (Section 3.3), latent interaction (Section 3.4), and output queries (Section 3.5). Notation is summarized in Appendix 18.

3.1 Token Construction

Atomizer-IO represents an input as a set of elementary observations, or *atoms*. Each atom holds a measured value $v \in \mathbb{R}$ (e.g. reflectance) and metadata $u_1, \ldots, u_J$ describing its acquisition. We combine these into tokens $\mathbf{f} = [\phi_0(v), \phi_1(u_1), \ldots, \phi_J(u_J)]$, where $\phi_0, \ldots, \phi_J$ are encoders for

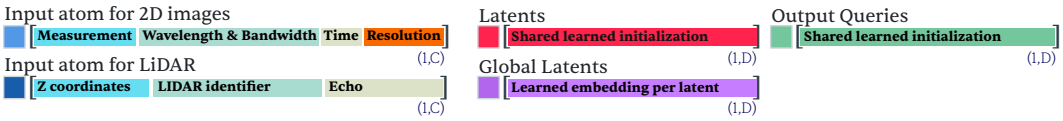


Figure 3: Token construction in Atomizer-IO. 2D imagery and LiDAR share the same atomic interface of measurements and acquisition metadata; spatial latents, global latents, and output queries use learned initializations.

each metadata field (e.g. wavelength and resolution). The input is represented as $\mathcal{F} = \{(\mathbf{f}_i, \mathbf{p}_i)\}_{i=1}^{N}$, pairing each token with its spatial position $\mathbf{p}_i \in \mathbb{R}^2$. Appendix B.4 further specifies our encoding schemes. To prevent the model from learning position-specific features, $\mathbf{p}_i$ is not included in $\mathbf{f}_i$ and is used only by the spatial sampler (Section 3.2). An atom's metadata fields follow from the sensor, and the architecture is indifferent to which ones it receives (Figure 3). For 2D rasters, v is the value measured at one pixel, in one spectral band, at one acquisition time, and the metadata describe that context: ground sampling distance, spectral support, and acquisition date. For 3D point clouds, v is the elevation, with a sensor identifier, the position of the return within the LiDAR pulse, and the return intensity when available as metadata. Nothing in the representation depends on a channel count or a grid, and atoms of different kinds can coexist in the same input set, as when LiDAR returns and co-registered image pixels are processed together.

3.2 Spatial Sampler

The spatial sampler (Figure 4) operates on two sets: a *query* set $\mathcal{Q} = \{(q_j, \mathbf{p}_j)\}_{j=1}^{|\mathcal{Q}|}$ seeking information from a *context* set $\mathcal{C} = \{(c_i, \mathbf{p}_i)\}_{i=1}^{|\mathcal{C}|}$. It proceeds in two steps: first, it establishes which context elements belong to the receptive field of each query; then, information flows from each receptive field to its query. In the encoder, input atoms form the context set and spatial latents the query set; in the decoder, spatial latents form the context set and output locations the query set.

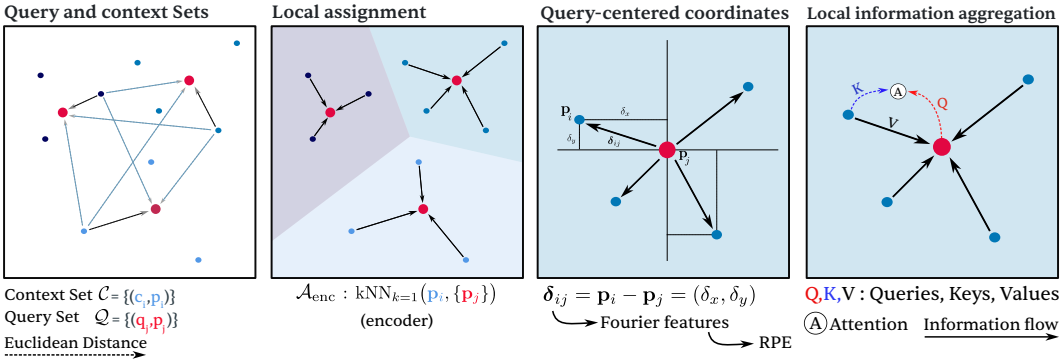


Figure 4: Spatial sampler. From left to right: query and context sets; construction of a local receptive field from spatial proximity, illustrated by the encoder's nearest-latent assignment and resulting Voronoi partition; query-centered relative coordinates; local cross-attention.

Assignment. We denote by $\mathcal{A}(\mathcal{C}, \mathcal{Q}) = \{\mathcal{V}_j\}_{j=1}^{|\mathcal{Q}|}$ the spatial assignment, where $\mathcal{V}_j$ is the index set of context elements in the local receptive field of query q_j. Receptive fields are built by nearest-neighbor search in the horizontal plane, which suits bird's-eye-view reasoning in Earth observation and applies to both rasters and 3D point clouds. In the encoder, each context element is assigned to its nearest query, inducing a Voronoi partition, which we denote by $\mathcal{A}_{\text{enc}}$. In the decoder, each query instead gathers its k nearest context elements, which we denote by $\mathcal{A}_{\text{dec}}$. Implementation details are given in Appendix B.8.

Information flow. Cross-attention defines a one-way flow: each query gathers information from its receptive field, while context elements themselves are not updated. Geometry is therefore expressed from the perspective of the receiving query. For a context element $i \in \mathcal{V}_j$, we use the relative offset $\boldsymbol{\delta}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and encode it as $\phi_\delta(\boldsymbol{\delta}_{ij})$ using Fourier features, analogous to the positional encoding used in NeRF (Mildenhall et al., 2021). The encoded offset is concatenated to the context feature before information is passed to the query through cross-attention (Appendix B.5).

3.3 ENCODER

In the encoder, spatial latents form the query set and input atoms form the context set of the spatial sampler.

Adaptive latent allocation. No operation in the architecture requires a fixed number or layout of spatial latents. In practice, we scale the number L of spatial latents with the number of input observations and place them uniformly over the spatial extent of the input. The values of L and the layouts used in our experiments are given in Appendix D.2. Each latent has an anchor position $\mathbf{p}_\ell \in \mathbb{R}^2$ and shares the same learned initialization $\mathbf{h}_{\text{init}} \in \mathbb{R}^D$, where D is the latent embedding dimension. We denote the initialized latent set by $\mathcal{H}^{\text{init}} = \{(\mathbf{h}_{\text{init}}, \mathbf{p}_\ell)\}_{\ell=1}^{L}$. Figure 5 illustrates the one instance of a spatial partition on a DALES scene.

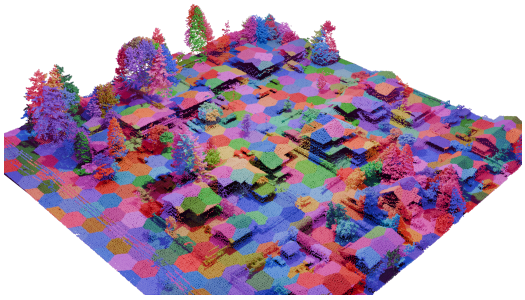


Figure 5: Example spatial latent partition on DALES (Section 4.4). Colors show the local receptive fields induced by assigning each input atom to its nearest spatial latent.

Local encoding. For the encoder, the spatial sampler returns the assigned atoms for each anchored latent $\{\mathcal{V}_\ell\}_{\ell=1}^{L} = \mathcal{A}_{\text{enc}}(\mathcal{F}, \mathcal{H}^{\text{init}})$. For an atom $i \in \mathcal{V}_\ell$, we compute the relative offset $\boldsymbol{\delta}_{i\ell} = \mathbf{p}_i - \mathbf{p}_\ell$, concatenate its Fourier encoding to the atom feature, and process the result with a shared MLP: $\mathbf{z}_{i\ell} = \text{MLP}([\mathbf{f}_i; \phi_\delta(\boldsymbol{\delta}_{i\ell})])$. Each spatial latent is then updated by local cross-attention over its processed context, $\mathbf{h}_\ell^{(0)} = \text{CrossAttn}_{\text{enc}}(\mathbf{h}_{\text{init}}, \{\mathbf{z}_{i\ell} : i \in \mathcal{V}_\ell\})$. The MLP and cross-attention weights are shared across latents, so the parameter count is independent of L. The resulting encoded latent set is $\mathcal{H}^{(0)} = \{(\mathbf{h}_\ell^{(0)}, \mathbf{p}_\ell)\}_{\ell=1}^{L}$.

For dense local contexts, we randomly sample at most m indices from each $\mathcal{V}_\ell$ before cross-attention, bounding the attention cost per latent. On Sen1Floods11 (Section 4.1), where inputs contain 15 channels at 512×512 resolution, a global cross-attention mechanism would need to discard 99.987% of the input atoms to match the same number of query–context interactions. Local sampling instead distributes this budget across the spatial partition and exposes the model to different subsets of each local context over training (Appendix B.7).

3.4 PROCESSOR

The processor operates on the encoded spatial latent set $\mathcal{H}^{(0)} = \{(\mathbf{h}_\ell^{(0)}, \mathbf{p}_\ell)\}_{\ell=1}^{L}$ together with G learned global latents $\mathcal{G}^{(0)} = \{\mathbf{g}_r^{(0)}\}_{r=1}^{G}$, with $\mathbf{g}_r^{(0)} \in \mathbb{R}^D$, similar in spirit to register tokens (Darcet et al., 2024). A processor of depth B stacks global multi-head self-attention blocks over the spatial and global latents, $(\mathcal{H}^{(b)}, \mathcal{G}^{(b)}) = \text{Block}_b(\mathcal{H}^{(b-1)}, \mathcal{G}^{(b-1)})$ for $b = 1, \ldots, B$, allowing information gathered from different local neighborhoods to propagate across the latent representation. The final processed spatial latent set is $\mathcal{H}^{(B)}$.

Latent spatial processing. While the encoder introduces atom–latent geometry through relative positional encoding in the spatial sampler, spatial relationships *between latents* are represented using two-dimensional rotary positional encoding (RoPE) (Heo et al., 2024) based on their anchor positions $\mathbf{p}_\ell$. Because the distance between anchors depends on how they are distributed over the input extent, no single compression scale suits every input; we therefore use a learned scale $s_p > 0$, initialized to a reference scale s_0 (Appendix B.6). Coordinates are compressed with s_p before applying RoPE to the queries and keys of the spatial latents. Global latents have no associated spatial position and are therefore left unrotated.

Temporal processing. Multitemporal inputs require no dedicated temporal module: atoms from all acquisitions can be encoded jointly in the same input set, with acquisition time provided as metadata. When explicit temporal aggregation is preferred, the encoder instead processes each timestep independently, with shared weights and identical spatial anchor positions, producing $\mathcal{H}_t^{(0)} = \{(\mathbf{h}_{t\ell}^{(0)}, \mathbf{p}_\ell)\}_{\ell=1}^{L}$ for each timestep t. The processor is then applied independently within each timestep, yielding $\mathcal{H}_t^{(B)} = \{(\mathbf{h}_{t\ell}^{(B)}, \mathbf{p}_\ell)\}_{\ell=1}^{L}$. For each spatial anchor ℓ, the T representations $\{\mathbf{h}_{t\ell}^{(B)}\}_{t=1}^{T}$ are aggregated along the temporal axis by a small transformer with a learned aggregation token, using temporal RoPE with the same coordinate compression and a dedicated

learned scale. This produces a temporally aggregated representation $\bar{\mathbf{h}}_\ell$ for each anchor, defining $\bar{\mathcal{H}}^{(B)} = \{(\bar{\mathbf{h}}_\ell, \mathbf{p}_\ell)\}_{\ell=1}^{L}$. When temporal aggregation is used, this set replaces $\mathcal{H}^{(B)}$ as decoder context. Its contribution is evaluated in Appendix B.1.

3.5 Decoder

The decoder predicts at a set of requested spatial positions $\{\mathbf{p}_j\}_{j=1}^{M}$. Each output query shares the same learned initialization $\mathbf{q}_{\text{init}} \in \mathbb{R}^D$, giving the initialized query set $\mathcal{Q}^{\text{init}} = \{(\mathbf{q}_{\text{init}}, \mathbf{p}_j)\}_{j=1}^{M}$. Below, $\mathcal{H}^{(B)}$ denotes the decoder context, replaced by $\bar{\mathcal{H}}^{(B)}$ when explicit temporal aggregation is used. The spatial sampler returns $\{\mathcal{V}_j\}_{j=1}^{M} = \mathcal{A}_{\text{dec}}(\mathcal{H}^{(B)}, \mathcal{Q}^{\text{init}})$. For a latent $\ell \in \mathcal{V}_j$, we compute the relative offset $\boldsymbol{\delta}_{\ell j} = \mathbf{p}_\ell - \mathbf{p}_j$ and concatenate its Fourier encoding to the latent feature. Each output query is then obtained by local cross-attention over this context, $\mathbf{q}_j = \text{CrossAttn}_{\text{dec}}(\mathbf{q}_{\text{init}}, \{[\mathbf{h}_\ell^{(B)}; \phi_\delta(\boldsymbol{\delta}_{\ell j})] : \ell \in \mathcal{V}_j\})$, and passed to a task-specific prediction head; for dense prediction, the input $\mathbf{q}_{\text{init}}$ is replaced by the input-conditioned query $\tilde{\mathbf{q}}_j$ defined below. Because output locations are queried explicitly, the decoder does not need to evaluate every spatial position uniformly. We exploit this property in Section 4.3.

For dense prediction, we additionally condition each output query on the observations at its target location before it attends to the latent representation, a local pathway analogous to the skip connections of U-Net (Ronneberger et al., 2015). Let $\mathcal{S}_j \subseteq \{1, \ldots, N\}$ be the indices of atoms co-located with query j: the atoms at the pixel being predicted for a 2D raster, and the atoms at the queried point for a point cloud. The query is first updated by cross-attention over their features, $\tilde{\mathbf{q}}_j = \text{CrossAttn}_{\text{obs}}(\mathbf{q}_{\text{init}}, \{\mathbf{f}_i : i \in \mathcal{S}_j\})$, combining observations at the target location with the processed information its surroundings provide through the latent cross-attention above.

4 Experiments

We evaluate Atomizer-IO by progressively relaxing the assumptions of a regular grid. We begin with conventional raster Earth observation tasks, then vary input composition and output density, remove the input grid entirely with irregular 3D point clouds, and finally test spatial reasoning under unseen input geometries.

4.1 Performance Across Diverse EO Tasks

Architectures that relax fixed input structure for greater generality may give up useful task- or sensor-specific priors, requiring more of the task structure to be learned from data. Before relaxing the grid, we therefore verify that Atomizer-IO is competitive when it is fully present, across a deliberately heterogeneous suite of eight datasets spanning different sensing modalities, channel counts, temporal lengths, spatial resolutions, input sizes, and prediction problems (Table 1). Each dataset is treated as an independent task and a separate model is trained from scratch.

Table 1: Datasets used for evaluation, spanning raster Earth observation and irregular LiDAR inputs. **T**: timesteps; **C**: input channels; **Geometry**: spatial structure of the input; **GSD**: ground sampling distance, undefined for point clouds; **Out. C**: output classes or regression targets.

Dataset	Sensor	T	C	Geometry	GSD	Problem	Out. C
ForestNet (Irvin et al., 2020)	Landsat-8	1	6	332^2	15 m	cls	12
BurnScars (Jakubik et al., 2023)	HLS	1	6	512^2	30 m	seg	2
EuroSAT (Helber et al., 2019)	Sentinel-2	1	13	64^2	10 m	cls	10
Cashew (Jin et al., 2021)	Sentinel-2	1	13	256^2	10 m	seg	7
Sen1Floods11 (Rambour et al., 2020)	S1 + S2	1	15	512^2	10 m	seg	2
xView2 (Gupta et al., 2019)	RGB (VHR)	2	3	512^2	0.5 m	seg	5
BioMassters (Nascetti et al., 2023)	S1 + S2	3	14	256^2	10 m	reg	1
PASTIS (Garnot & Landrieu, 2021)	S1 + S2	6	10	128^2	10 m	seg	19
DALES (Varney et al., 2020)	LiDAR	1	1	point set	–	3D seg	8
FRACTAL (Gaydon et al., 2024)	LiDAR + VHR	1	4	points + image	–	3D seg	7

We compare Atomizer-IO with grid-based baselines (ResNet50 (He et al., 2016) and ViT), flexible EO-specific models (RAMEN and UniverSat), and Perceiver-IO as a generic set-based latent architecture. All models are trained from scratch under matched conditions (Appendix D.2).

Table 2: Single-task performance. Best in **bold**, second best underlined. Classification uses macro F1, segmentation mIoU, and BioMassters RMSE (Mg/ha, ↓).

Model	EuroSAT	ForestNet	BurnScars	Sen1Floods	PASTIS	xView2	Cashew	BioMassters ↓
ResNet50	90.9	43.8	82.8	87.8	30.0	53.8	69.9	46.3
ViT	90.7	39.1	87.5	85.0	35.1	52.4	47.5	52.7
RAMEN	**92.1**	40.4	88.2	89.1	**45.5**	53.3	64.2	44.7
UniverSat	90.8	**44.3**	88.1	86.8	42.9	**56.3**	62.4	46.0
Perceiver-IO	90.9	35.4	83.4	92.2	17.9	43.8	26.2	52.2
Atomizer-IO	90.2	35.8	**88.8**	**93.2**	44.1	54.6	**72.0**	**42.4**

Across the eight tasks, Atomizer-IO excels at spatially-explicit tasks, ranking first on four and second on two (Table 2), matching or exceeding grid-based and flexible EO architectures on their own terms, on raster data. This establishes the baseline for the relaxations that follow. These tasks rely on different sources of information. PASTIS benefits strongly from reasoning across acquisitions, Sen1Floods11 from local spectral and radar measurements, while Cashew shows a stronger dependence on spatial context. Appendix B examines these differences directly by ablating temporal aggregation, spatial reasoning, and local observation conditioning across representative tasks. As expected, we also observe that Atomizer-IO does not have an edge over Perceiver-IO for non spatially explicit classification tasks, such as EuroSAT and ForestNet.

4.2 Flexible Input Composition: Handling Missing Channels

In Earth observation, available channels can change with sensor availability, weather conditions or spectral coverage. Models built around a fixed channel layout must preserve that layout when observations are missing, whereas flexible representations can omit unavailable observations directly. We evaluate this setting on Sen1Floods11 and BioMassters, both combining Sentinel-1 and Sentinel-2 observations. In this experiment, all models use the same band-dropout augmentation, so variation in channel availability is anticipated during training. Atomizer-IO and UniverSat omit unavailable observations, whereas fixed-channel models are zero-padded at the corresponding positions. The same trained model is evaluated under six input configurations, from the full observation set to single-sensor and restricted spectral subsets.

Table 3: Performance under flexible input composition on Sen1Floods11 (mIoU, %) and BioMassters (RMSE, Mg/ha). All models are trained under the same band-dropout protocol.

	Sen1Floods11 ↑						**BioMassters ↓**					
Test Configuration	**Atom.**	**Res.**	**ViT**	**Perc.**	**RAM.**	**Uni.**	**Atom.**	**Res.**	**ViT**	**Perc.**	**RAM.**	**Uni.**
All bands	**92.9**	88.1	84.5	92.1	87.0	88.7	**41.5**	47.6	52.9	52.0	43.4	45.7
S2 only	**92.8**	86.4	84.4	92.2	87.8	88.6	**45.8**	55.9	55.5	57.7	48.4	49.1
S1 only	**80.0**	66.4	75.3	77.2	75.1	79.4	**48.1**	52.1	55.5	58.6	49.2	49.1
RGB only	**44.8**	43.7	43.8	43.8	44.5	43.8	71.0	73.8	71.4	**70.4**	71.9	70.8
No SWIR	**87.8**	49.0	75.1	81.8	63.7	85.2	**43.0**	49.1	54.0	53.4	46.0	46.7
No red-edge	**92.9**	85.3	80.5	89.1	85.9	88.0	**42.2**	49.8	53.8	54.0	45.4	46.4
Average	**81.9**	69.8	73.9	79.4	74.0	78.9	**48.6**	54.7	57.2	57.7	50.7	51.3

Atom.: Atomizer-IO; Res.: ResNet50; Perc.: Perceiver-IO; RAM.: RAMEN; Uni.: UniverSat.

Table 3 shows that Atomizer-IO achieves the best average performance on both datasets and leads in most configurations. Its advantage is largest when specific bands or an entire sensor are removed, but nearly disappears under RGB only, where much of the discriminative information is lost. More broadly, architectures that relax the fixed channel layout tend to handle changes in input composition better than conventional fixed-layout baselines, supporting the value of decoupling representation from a prescribed channel structure.

4.3 Flexible Output Decoding: Inference-Time Compute Control

Here we study various relaxations of the output grid. Atomizer-IO predicts at explicitly requested spatial locations, so dense segmentation corresponds simply to placing one query at every pixel (*Dense* decoding). Because the decoder does not require all locations to be queried uniformly, output density can instead be adapted to the spatial complexity of the prediction.

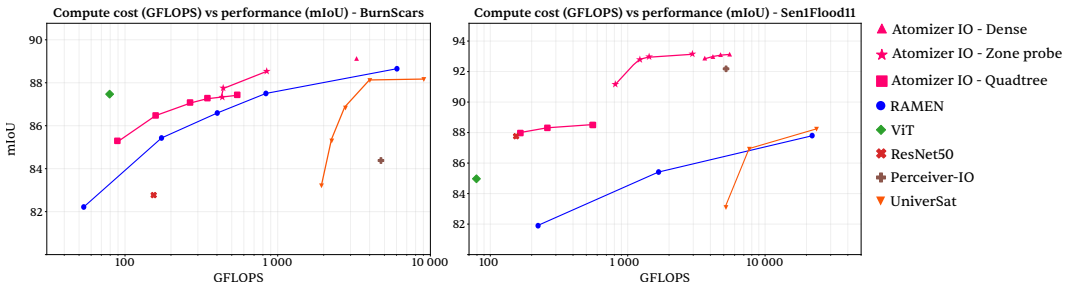


Figure 6: Compute cost (GFLOPs) versus segmentation performance (mIoU). Dense, Zone-probe, and Quadtree decoding vary inference cost through different output query strategies, using the same Atomizer-IO checkpoint without retraining.

We exercise this property by approximating dense predictions while querying fewer locations. For each region, we first probe a small number of locations. If all probes predict the same class, that class is assigned to the remaining locations without querying them; otherwise, additional computation is allocated to the region. *Zone-probe* applies this rule to regions associated with individual latents, while *Quadtree* recursively subdivides disagreeing regions and probes them at finer spatial scales. All strategies use the same trained checkpoint and differ only in how densely the output space is queried. Details and complexity are given in Appendix C.1. Figure 6 shows that a single trained checkpoint spans a broad range of inference costs, with *Dense* decoding being competitive with the flexible EO baselines at the high-compute end. The two adaptive strategies reduce cost differently: *Zone-probe* couples output regions to the model's latent partition, while *Quadtree* adapts output density independently of that partition through recursive refinement.

Atomizer-IO therefore exposes complementary inference-time controls: like RAMEN and UniverSat, compute can be reduced through a coarser latent representation, while the query-based decoder additionally allows output density to vary after training. Probe agreement is a heuristic rather than a guarantee, reflecting the trade-off between exact dense evaluation and reduced inference cost.

4.4 Flexible Input Geometry: Unordered LiDAR Point Clouds

We break free of the input grid for semantic segmentation of 3D point clouds. Our goal is not to outperform 3D-specific architectures, but to demonstrate the flexibility of our framework. On DALES and FRACTAL (Table 1), each LiDAR point becomes an atom (Appendix D). On FRACTAL, RGB pixels from co-registered orthophotos are additionally encoded as atoms, allowing both modalities to be jointly processed. Training and inference tiling, point sampling, and batching details are given in Appendix D.4.

Table 4: DALES and FRACTAL segmentation results (mIoU, %).

DALES	
Method	**mIoU**
KPConv (Thomas et al., 2019)	**81.1**
SPT (Robert et al., 2023)	79.6
PointNet++ (Qi et al., 2017b)	68.3
ConvPoint (Boulch, 2020)	67.4
Atomizer-IO	66.7
SPG (Landrieu & Simonovsky, 2018)	60.6
PointCNN (Li et al., 2018)	58.4
ShellNet (Zhang et al., 2019)	57.4
Perceiver-IO (Jaegle et al., 2021a)	48.7
FRACTAL	
Atomizer-IO	**78.4**
Perceiver-IO (Jaegle et al., 2021a)	78.1
RandLA-Net (Hu et al., 2020)	77.5

Atomizer-IO is competitive on both datasets without 3D-specific design beyond its tokenizer, ahead of three of the DALES 3D baselines and best on FRACTAL (Table 4). Perceiver-IO performs poorly on DALES, highlighting the importance of geometric structure in the latent reasoning, but both models outperform RandLA-Net on FRACTAL, suggesting the image modality alone may already be a strong predictor for this benchmark. Our gap to KPConv and SPT on DALES suggests opportunities for geometry-adaptive 3D latent partitions, hierarchical reasoning, and finer local 3D modeling. We leave these extensions to future work.

4.5 Controlled Analysis of Spatial Structure

Table 5: RoPE ablation. Accuracy on MNIST and mIoU on Sen1Floods11.

	MNIST	Sen1Fl11
No RoPE	29.48	92.42
RoPE	**99.37**	**93.17**

Removing the input grid does not remove the need for spatial structure. In remote-sensing tasks, however, this is difficult to isolate because individual observations can already be highly informative: a local spectral signature may indicate water, vegetation, or other materials without requiring much spatial context. We therefore use MNIST as a controlled setting. Because MNIST images are nearly binary, individual pixel intensities carry little information in isolation, and classification depends primarily on their spatial arrangement.

Atomizer-IO's spatial latents share the same learned initialization and attention weights and are distinguished by the observations they aggregate and their relative geometry. On MNIST, each latent carries only local intensity information; removing RoPE therefore prevents the processor from recovering their spatial arrangement and reduces accuracy from 99.37 to 29.48. On Sen1Floods11, in contrast, removing RoPE only reduces mIoU from 93.17 to 92.42, consistent with the strong predictive content already present in the local multispectral and radar observations. Broader ablations in Appendix B further examine how different tasks rely on local observation content, spatial reasoning, and temporal structure.

We finally test whether the same model can process input geometries not seen during training. Atomizer-IO and a parameter-matched ViT with 1×1 pixel patches are trained only on complete MNIST images. At test time, the same background pixels are removed entirely from both models while all digit pixels are retained, so both receive exactly the same surviving observations. As the background becomes increasingly sparse and irregular, Atomizer-IO degrades more gracefully, retaining 94.93% accuracy when only 25% of the background pixels are kept, compared with 72.17% for ViT (Table 6). This suggests that Atomizer-IO better preserves its ability to exploit task-relevant observations under substantial changes in input geometry.

Table 6: MNIST accuracy (%) under unseen input geometries. Digit pixels are always retained; both models use the same checkpoint trained on complete inputs.

Background kept (%)	**Atomizer-IO**	**ViT**
100	**99.26**	98.11
75	**98.72**	97.65
50	**98.37**	94.85
25	**94.93**	72.17
10	**74.36**	37.24

5 Conclusion

Computing on a regular grid is not required for competitive performance under a competitive compute budget. Across eight Earth-observation benchmarks and two LiDAR datasets spanning classification, regression, and 2D/3D segmentation, Atomizer-IO remains competitive with grid-based and flexible EO architectures while progressively relaxing assumptions on input and output structure. To our knowledge, no other EO architecture has been evaluated on both multispectral rasters and unordered LiDAR point clouds through the same encoder and without modality-specific processing branches.

Unlike architectures whose structure reflects assumptions about the data, Atomizer-IO encodes sensing information explicitly in its input tokens. Each atom represents a measurement at a particular place, time, and sensor, without requiring a fixed shape, channel count, or neighboring measurement; locality and geometry are introduced afterwards from physical relationships between observations.

As future work, we expect this perspective to be particularly relevant to large-scale pretraining, where a central challenge is reconciling observations that differ in what they measure and how they are sampled. Existing architectures typically begin from a tensor representation and introduce dedicated mechanisms to accommodate each axis of variation. Atomizer-IO starts from the opposite premise: the observation itself is the common unit. The results in this paper, in which the exact same architecture provides competitive results across a wide range of modalities, suggest that Atomizer-IO could make a good candidate architecture for a multi-modal EO foundation model.

Our results allow us to conclude that pixels, patches, and grids need not define the interface of a sensing architecture. They can instead be treated as particular discretizations of observations, introduced when useful rather than assumed from the start.

AI USE STATEMENT

In this work, we used generative AI tools to implement the methods, correct grammar, and edit the research paper to improve readability. We have not used generative AI tools to generate synthetic data sets, develop theoretical models, formulate or prove mathematical claims, propose hypotheses, design experiments, translate text, clean data sets, or interpret results, and all other required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. Specifically, all AI-generated code and text suggestions were thoroughly reviewed, manually verified for correctness, and substantially modified by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Anysat: An earth observation model for any resolutions, scales, and modalities. *arXiv preprint arXiv:2412.14123*, 2024.

Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24–34, 2020.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Hugo Riffaud de Turckheim, Diego Marcos, Roberto Interdonato, and Sylvain Lobry. Atomizer: Generalizing to unseen modalities by breaking images down to a set of scalars. In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA, 2025. URL https://bmva-archive.org.uk/bmvc/2025/assets/papers/Paper_1058/paper.pdf.

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

Vivien Sainte Fare Garnot and Loic Landrieu. Lightweight temporal self-attention for classifying satellite image time series, 2020. URL https://arxiv.org/abs/2007.00586.

Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4852–4861. IEEE, 2021.

Charles Gaydon, Michel Daab, and Floryne Roche. Fractal: An ultra-large-scale aerial lidar dataset for 3d semantic segmentation of diverse landscapes. *arXiv preprint arXiv:2405.04634*, 2024.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *arXiv preprint arXiv:1912.12033*, 2019.

Ritwik Gupta, Richard Hosfelt, Sandra Sajeev, Nirav Patel, Bryce Goodman, Jigar Doshi, Eric Heim, Howie Choset, and Matthew Gaston. xbd: A dataset for assessing building damage from satellite imagery. *arXiv preprint arXiv:1911.09296*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Patrick Helber, Benjamin Bischke, Andreas 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(7):2217–2226, 2019.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pp. 289–305. Springer, 2024.

Nicolas Houdré, Diego Marcos, Hugo Riffaud de Turckheim, Dino Ienco, Laurent Wendling, Camille Kurtz, and Sylvain Lobry. Ramen: Resolution-adjustable multimodal encoder for earth observation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27838–27848, June 2026.

Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11105–11114. IEEE, 2020.

Jeremy Irvin, Hao Sheng, Neel Ramachandran, Sonja Johnson-Yu, Sharon Zhou, Kyle Story, Rose Rustowicz, Cooper Elsworth, Kemen Austin, and Andrew Y Ng. Forestnet: Classifying drivers of deforestation in indonesia using deep learning on satellite imagery. *arXiv preprint arXiv:2011.05479*, 2020.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021a.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021b.

Johannes Jakubik, Sujit Roy, CE Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, et al. Foundation models for generalist geospatial artificial intelligence. *arXiv preprint arXiv:2310.18660*, 2023.

Johannes Jakubik, Felix Yang, Benedikt Blumenstiel, Erik Scheurer, Rocco Sedona, Stefano Maurogiovanni, Jente Bosmans, Nikolaos Dionelis, Valerio Marsocci, Niklas Kopp, et al. Terramind: Large-scale generative multimodality for earth observation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7383–7394. IEEE, 2025.

Z Jin, C Lin, C Weigl, J Obarowski, and D Hale. Smallholder cashew plantations in benin. *Radiant MKHub*, 2021.

Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Xuyang Li, Chenyu Li, Pedram Ghamisi, and Danfeng Hong. Fleximo: A flexible remote sensing foundation model. *arXiv preprint arXiv:2503.23844*, 2025.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

Valerio Marsocci, Yuru Jia, Georges Le Bellier, David Kerekes, Liang Zeng, Sebastian Hafner, Sebastian Gerard, Eric Brune, Ritu Yadav, Ali Shibli, et al. Pangaea: A global and inclusive benchmark for geospatial foundation models. *arXiv preprint arXiv:2412.04204*, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Andrea Nascetti, Ritu Yadav, Kirill Brodt, Qixun Qu, Hongwei Fan, Yuri Shendryk, Isha Shah, and Christine Chung. Biomassters: A benchmark dataset for forest biomass estimation using multi-modal satellite time-series. *Advances in Neural Information Processing Systems*, 36:20409–20420, 2023.

Yohann Perron, Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Universat: Resolution- and modality-agnostic transformers for earth observation. In *Advances in Neural Information Processing Systems*, 2026. To appear.

Jonathan Prexl and Michael Schmitt. Senpa-mae: Sensor parameter aware masked autoencoder for multi-satellite self-supervised pretraining. In *DAGM German Conference on Pattern Recognition*, pp. 317–331. Springer, 2024.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Clément Rambour, Nicolas Audebert, E Koeniguer, Bertrand Le Saux, Michel Crucianu, and Mihai Datcu. Flood detection in time series of optical and sar images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43(B2):1343–1346, 2020.

Colorado J Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4088–4099, 2023.

Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3d semantic segmentation with superpoint transformer. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Gencer Sumbul, Chang Xu, Emanuele Dalsasso, and Devis Tuia. Smarties: Spectrum-aware multi-sensor auto-encoder for remote sensing images. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5569–5578. IEEE, 2025.

Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, Orsteinn Elí Gíslason, Benedikt Blumenstiel, Rinki Ghosal, Pedro Henrique De Oliveira, Joao Lucas de Sousa Almeida, Rocco Sedona, Yanghui Kang, et al. Prithvi-eo-2.0: A versatile multi-temporal foundation model for earth observation applications. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

Gabriel Tseng, Ruben Cartuyvels, Ivan Zvonkov, Mirali Purohit, David Rolnick, and Hannah Kerner. Lightweight, pre-trained transformers for remote sensing timeseries. *arXiv preprint arXiv:2304.14065*, 2023.

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 and local features in pretrained remote sensing models. *arXiv preprint arXiv:2502.09356*, 2025.

Nina Varney, Vijayan K Asari, and Quinn Graehling. Dales: A large-scale aerial lidar data set for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 717–726. IEEE, 2020.

Leonard Waldmann, Ando Shah, Yi Wang, Nils Lehmann, Adam Stewart, Zhitong Xiong, Xiao Xiang Zhu, Stefan Bauer, and John Chuang. Panopticon: Advancing any-sensor foundation models for earth observation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2204–2214, 2025.

Ben G Weinstein, Sarah J Graves, Sergio Marconi, Aditya Singh, Alina Zare, Dylan Stewart, Stephanie A Bohlman, and Ethan P White. A benchmark dataset for canopy crown detection and delineation in co-registered airborne rgb, lidar and hyperspectral imagery from the national ecological observation network. *PLoS computational biology*, 17(7):e1009180, 2021.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European conference on computer vision*, pp. 432–448. Springer, 2018.

Zhitong Xiong, Yi Wang, Fahong Zhang, Adam J Stewart, Joëlle Hanna, Damian Borth, Ioannis Papoutsis, Bertrand Le Saux, Gustau Camps-Valls, and Xiao Xiang Zhu. Neural plasticity-inspired multimodal foundation model for earth observation. *arXiv preprint arXiv:2403.15356*, 2024.

Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1607–1616. IEEE, 2019.