

Maya4: a Novel Large-Scale Multi-Level SAR Representation Dataset for Machine Learning

Roberto Del Prete^a, Gabriele Daga^a, Cédric Léonard^{b,c}, Georgia Channing^d, and Nicolas Longépé^a

^aESA Φ-lab, European Space Research Institute (ESRIN), Frascati, Italy

^bTechnical University of Munich, Munich, Germany

^cRemote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany

^dHugging Face, SAS, Paris, France

Abstract

Maya4 is a cloud-native Sentinel-1 Stripmap SAR dataset that exposes four aligned processing states: raw, rc, rcmc, and focused az. Unlike existing public resources limited to focused imagery, Maya4 exposes the signal chain from Level-0 echoes to focused products, enabling operator-level learning and evaluation. The release contains about 2 TB of data with chunked/sharded storage and co-located metadata for scalable access. Maya4 supports three broad classes of signal-domain use cases: compression, learned focusing, and cross-level representation learning, together with benchmarking against conventional processors. The dataset is publicly available at <https://huggingface.co/buckets/ESA-philab/Maya4>.

1 Introduction

Synthetic Aperture Radar (SAR) provides coherent microwave imaging independent of daylight and robust to most weather conditions [1]. By integrating echoes along the flight path, it synthesizes a long aperture and achieves high range and azimuth resolution [1, 2].

Many widely used open SAR resources primarily expose focused Level-1 products. Public access to intermediate focusing states—including the raw-domain, range-compressed, and migration-corrected representations considered in this work—remains comparatively limited, which constrains Machine Learning (ML) methods to image-domain supervision and obscures the operator-level transformations that produce Single Look Complex (SLC) data [1].

Modern deep-learning models are fundamentally data-hungry: robust training requires large sample volumes, broad scene diversity, and supervision signals that remain consistent across acquisition conditions. When training data are restricted to final focused products, models tend to learn task-specific image statistics rather than reusable signal-domain structure, and their generalization across sensors, geometries, or downstream objectives can degrade. Providing intermediate processing states therefore directly improves the quality of supervision for representation learning, inverse modeling, and end-to-end optimization in radar processing pipelines.

Maya4 addresses this gap by releasing Sentinel-1 Stripmap acquisitions at four processing stages used throughout this paper: raw, rc, rcmc, and focused az. By making these aligned intermediate representations available within a single dataset, Maya4 supports signal-domain learning and evaluation settings that are developed more concretely in Section 4. The remainder of the paper is organized as follows: Section 2 positions Maya4 against existing benchmarks. Section 3 specifies the processing levels, production pipeline, and storage layout. Section 4 details representative ML use cases, and Section 5 summarizes the contribution.

2 Motivation & Related Work

Modern ML approaches in Earth Observation (EO) increasingly rely on extensive, well-curated, and labelled datasets. While optical remote-sensing research has advanced through numerous open resources, SAR data remain comparatively underrepresented. This imbalance arises from the intrinsic complexity of SAR signals—complex-valued, multi-polarised, and high-dimensional—as well as from operational and commercial constraints that limit access to raw or intermediate products. Consequently, most ML research on SAR has concentrated on final backscatter images or derived geophysical products, leaving the earlier stages of the signal chain largely unexplored [3–5].

Table 1 summarizes representative open SAR datasets, their released data representations, and the learning setups they primarily support, highlighting that Maya4 is the only public dataset exposing aligned raw, rc, rcmc, and az levels.

Recent open SAR benchmarks have primarily focused on downstream semantic tasks built on focused Sentinel products. SEN12MS and the DFC2020 benchmark target land-cover mapping from co-registered Sentinel-1/2 image patches and weakly supervised labels [6, 7]; SEN12-FLOOD targets flood-event detection from multi-modal Sentinel time series [8]; OpenSARShip supports maritime target interpretation with Sentinel-1 ship chips and associated metadata [9]; and Sen1Floods11 supports flood-water segmentation from annotated Sentinel-1 imagery [10]. In the object-detection domain, HRSID, SSDD, and SARDet-100K provide focused SAR scenes/chips with bounding-box annotations [11–13]. Unlike SEN12MS/DFC2020, SEN12-FLOOD, OpenSARShip, and Sen1Floods11, which are designed primarily for downstream semantic tasks on focused Sentinel products, Maya4 exposes multiple intermediate stages of the SAR focusing chain, enabling research on learned or alternative SAR processing operators.

The Maya4 dataset addresses this gap by exposing multiple intermediate representations of the SAR focusing pipeline. These intermediate layers—typically discarded—carry valuable statistical and structural information that can be ex-

1g.

Table 1 Comparison with representative open SAR datasets, including typical training policy and released data representation. Maya4 is the only public dataset releasing aligned *raw*, *rc*, *rcmc*, and focused *az* levels.

Dataset	Tasks	Training policy	Released representation	Temporal setup
SEN12MS [6]	Land-cover mapping and multimodal S1/S2 fusion	Weakly supervised downstream learning	Focused Sentinel-1/2 image patches (image domain)	Multi-season patch pairs
DFC2020 [7]	Weakly supervised global land-cover mapping benchmark	Weakly supervised challenge learning	Focused Sentinel-1/2 contest patches (image domain)	Predominantly single-date challenge snapshots
SEN12-FLOOD [8]	Flood-event detection	Supervised event detection	Focused co-registered Sentinel-1/2 imagery (image domain)	Event-centric time series
OpenSARShip [9]	Ship interpretation and detection	Supervised classification/detection	Focused Sentinel-1 ship chips (image domain)	Single-scene ship chips
Sen1Floods11 [10]	Flood-water segmentation	Supervised segmentation	Focused Sentinel-1 image chips with masks (image domain)	Multi-event, patch-based snapshots
HRSID [11]	High-resolution ship detection	Supervised detection	Focused SAR image scenes with ship bounding boxes (image domain)	Predominantly single-date scenes
SSDD [12]	Ship detection benchmark development	Supervised detection	Focused SAR image chips with ship annotations (image domain)	Predominantly single-scene chips
SARDet-100K [13]	Large-scale SAR object detection	Supervised detection	Focused SAR intensity scenes with object boxes (image domain)	Mostly single-date scene snapshots
Maya4 (ours)	Learned focusing, compression, and cross-level representation learning	Supervised, self-supervised, and hybrid physics-informed learning	Hierarchical SAR signal-chain states (<i>raw</i>, <i>rc</i>, <i>rcmc</i>, <i>az</i>) in signal and image domains	Per-acquisition multi-level access

Access to such multi-level signal states directly enables several emerging research directions: learned and task-adaptive compression of raw data, deep autoencoder-based SAR focusing, self-supervised representation learning from unfocused echoes, and physics-informed models that integrate knowledge of radar signal formation. Through this design, Maya4 not only complements existing SAR benchmarks but also bridges the gap between physical modelling and data-driven understanding of radar imaging.

3 Dataset Description

3.1 Processing Levels

Maya4 releases four checkpoints of the Sentinel-1 Stripmap focusing chain: *raw*, *rc*, *rcmc*, and *az* (Figure 1). We use $s[n, m]$ for range–azimuth signals, $S[k, m]$ for range-frequency representations, $S[n, \ell]$ for azimuth-frequency representations, and $S[k, \ell]$ for 2-D spectra, where n is range-sample index, m azimuth-line index, k range-frequency bin, and ℓ azimuth-frequency bin [1, 14–16]. The chain maps $s_{\text{raw}}[n, m]$ to focused $s_{\text{az}}[n, m]$.

- **Raw** (*raw*):

$$\begin{aligned}
 s_{\text{raw}}[n, m] = & \sum_{p=1}^P \sigma_p \\
 & \times \text{rect}\left(\frac{nT_s - 2R_p[m]/c}{T_p}\right) \\
 & \times \exp\left\{j\pi K_r \left(nT_s - \frac{2R_p[m]}{c}\right)^2\right\} \\
 & \times \exp\left\{-j\frac{4\pi f_c}{c} R_p[m]\right\}.
 \end{aligned} \quad (1)$$

where T_s is the range sampling interval, T_a the azimuth sampling interval, T_p the pulse duration, K_r the transmitted chirp rate, $R_p[m]$ the slant range to scatterer p at azimuth sample m , and σ_p its complex reflectivity. This stage preserves the full phase history and is the foundation for any synthetic aperture processing.

- **Range Compressed** (*rc*): Range compression is achieved by matched filtering along the range-sa . . . the Sentinel-1 Level-1 flow, range lines

and compressed through FFT-domain matched filtering:

$$S_{\text{raw}}[k, m] = \text{FFT}_r\{s_{\text{raw}}[n, m]\}, \quad (2)$$

$$S_{\text{rc}}[k, m] = S_{\text{raw}}[k, m] H_r^*[k], \quad (3)$$

$$s_{\text{rc}}[n, m] = \text{IFFT}_r\{S_{\text{rc}}[k, m]\}. \quad (4)$$

where $\text{FFT}_r\{\cdot\}$ and $\text{IFFT}_r\{\cdot\}$ denote the forward and inverse discrete Fourier transforms along the range index n , and $H_r[k]$ is the discrete spectral representation of the matched filter. This step increases range resolution by compressing the chirped pulse into a narrow impulse response, isolating targets by delay (range).

- **Range Cell Migration Corrected** (*rcmc*): During acquisition, platform motion causes scatterers at different ranges to migrate across range bins as m evolves. After azimuth FFT, the *rcmc* step is applied in the Range–Doppler domain:

$$S_{\text{rd}}[n, \ell] = \text{FFT}_a\{s_{\text{rc}}[n, m]\}, \quad (5)$$

followed by

$$S_{\text{rcmc}}[n, \ell] = \mathcal{I}_{\text{sinc}}(S_{\text{rd}}[\cdot, \ell], \Delta n_{\text{rcmc}}[n, \ell]), \quad (6)$$

where $\Delta n_{\text{rcmc}}[n, \ell]$ is the migration offset measured relative to the nominal slant range $R_0[n]$ of range cell n . In Maya4, $R_0[n]$ is computed from the Sentinel-1 slant-range time grid as $R_0[n] = \frac{c}{2}(\tau_0 + nT_s)$, where τ_0 is the slant-range time of the first range sample in the product metadata. This follows the Sentinel-1 Level-1 implementation, where range-cell migration correction is computed per range cell and applied via sinc-based range interpolation. After the *rcmc* step, the signal is geometrically stabilized and ready for azimuth compression.

- **Azimuth Focused** (*az*): The final stage focuses the signal in the azimuth-line dimension m using matched filtering in azimuth frequency:

$$s_{\text{az}}[n, m] = \text{IFFT}_a\{S_{\text{rcmc}}[n, \ell] H_{\text{az}}^*[\ell]\}, \quad (7)$$

where $\text{FFT}_a\{\cdot\}$ and $\text{IFFT}_a\{\cdot\}$ denote the forward and inverse discrete Fourier transforms along azimuth, and

$H_{\text{az}}[n, \ell] = \exp\left[-j\pi \frac{(f_\ell - f_{\text{dc}}[n])^2}{K_a[n]}\right]$ is a range-dependent filter parameterized by Doppler centroid

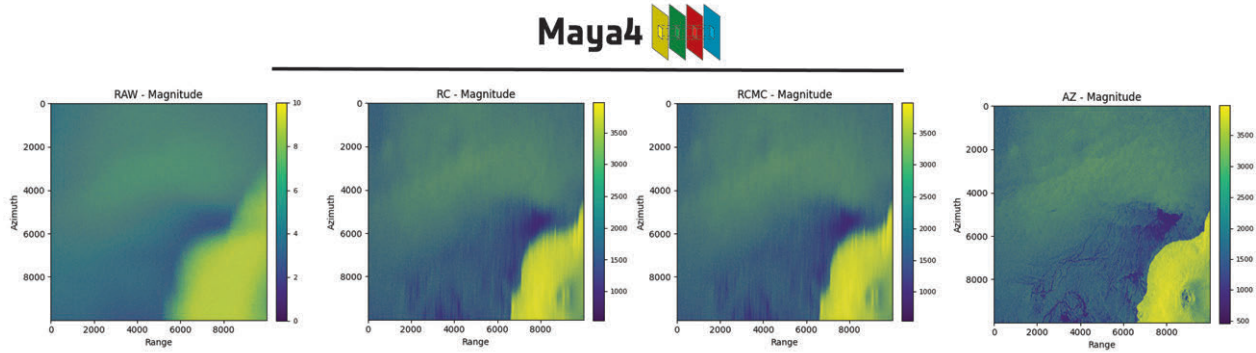


Figure 1 Maya4 processing stages: raw, rc, rcmc, and focused az.

$f_{dc}[n]$ and azimuth frequency-modulation (FM) rate $K_a[n]$. In Maya4, $f_{dc}[n]$ and $K_a[n]$ are derived from Sentinel-1 annotation metadata polynomials (with orbit/geometry terms from the product metadata), consistent with the Sentinel-1 Level-1 Detailed Algorithm Definition¹; data-driven Doppler-centroid estimators are considered as optional alternatives for experimental studies [16, 17]. This step coherently integrates the signal over the synthetic aperture, yielding a fully focused slant-range complex image $s_{az}[n, m]$ with high azimuth and range resolution.

To ensure full transparency and reproducibility of the processing chain, we provide an open-source reference implementation encompassing all transformation stages in `sarpyx` [18].

It is worth noting that the multi-level hierarchy captures a pragmatic discretization of the radar-focusing process from a raw time-domain signal to a geometrically and radiometrically focused image, rather than an exhaustive exposure of every internal operation. In Maya4, the released levels are four persisted checkpoints of that chain: raw, rc, rcmc, and az. By providing access to each signal state, Maya4 enables both the analysis of these transformations and the training of learned counterparts $\hat{C}_\theta$ that approximate them via machine learning.

3.2 Reference Processing Pipeline and Validation

Maya4 products are generated with `sarpyx` as the production toolchain [18]. Although `sarpyx` supports both a native Range-Doppler path and a SNAP-backed path, the released Maya4 intermediate levels are produced with the native RDA chain; the SNAP-backed flow is used for cross-checking focused outputs and metadata consistency.

The reference pipeline used to generate Maya4 is summarized as follows:

- **Raw packet decoding.** Sentinel-1 Level-0 packets are decoded into complex I/Q echo lines by parsing packet headers, timing counters, and ancillary telemetry, then assembling lines in acquisition order with associated orbit and attitude metadata.
- **Range reference function.** The range matched-filter reference is generated from Sentinel-1 transmitted chirp parameters available in metadata (including pulse duration, sampling interval, and chirp rate K_r), yielding the discrete $H_r[k]$ used in FFT-domain range compression.
- **FFT lengths and padding.** Range and azimuth transforms use zero-padded FFTs with configured lengths $N_{r,\text{fft}}$ and $N_{a,\text{fft}}$ (chosen to be no smaller than the valid sample sup-

port for each processing block); the same configuration is propagated across all levels of a product.

- **Secondary range compression.** Secondary range compression is not applied in the default Maya4 Stripmap production configuration; the processing uses the primary range matched-filter stage followed by the `rcmc` step and azimuth focusing.
- **rcmc interpolation kernel.** The `rcmc` step is performed in Range-Doppler coordinates with a sinc-based interpolation operator, consistent with Sentinel-1 Level-1 processing practice.
- **Doppler-centroid source.** Doppler centroid $f_{dc}[n]$ is taken from Sentinel-1 annotation polynomials; data-driven estimators (e.g., [17]) are treated as optional diagnostic alternatives.
- **Azimuth frequency-modulation (FM) rate source.** Azimuth frequency-modulation rate $K_a[n]$ is derived from Sentinel-1 annotation FM-rate polynomials together with acquisition geometry/orbit metadata, consistent with the mission algorithm definition [16].

3.3 Dataset Scope and Size

The dataset is divided into train, validation, and test subsets of Sentinel-1 Stripmap acquisitions (16/8/8 products). The total archive is roughly 2 TB and is stored in `Zarr` format, enabling scalable cloud-native access and efficient slicing by processing level, spatial tile, and temporal dimension. Figure 2 illustrates this split for the Europe-only acquisition footprint covered by Maya4 in April 2024. Marker shape indicates the subset, while marker color encodes acquisition date within that single-month window. The resulting distribution spans multiple regions across Europe and multiple acquisition dates within April 2024, supporting both spatial diversity and short-term temporal variability during model development and evaluation.

3.4 Data Format and Access

Maya4 is distributed as `Zarr` arrays to support scalable, cloud-native access (e.g., via `Dask` or `Xarray`). Each product is available at every released processing level, and each level can be read independently, so users can retrieve only the spatial or temporal subset required by a given experiment. Metadata (acquisition settings, processing state, and geolocation) are co-stored with the arrays to preserve reproducibility. All released `Zarr` groups are hosted in the public Hugging Face bucket `ESA-philab/Maya4` and distributed under the Apache License 2.0. This endpoint provides discoverability, release traceability, and straightforward HTTP-based programmatic access.

For efficient, Maya4 combines chunking and

¹See <https://sentiwiki.com/document-library>.

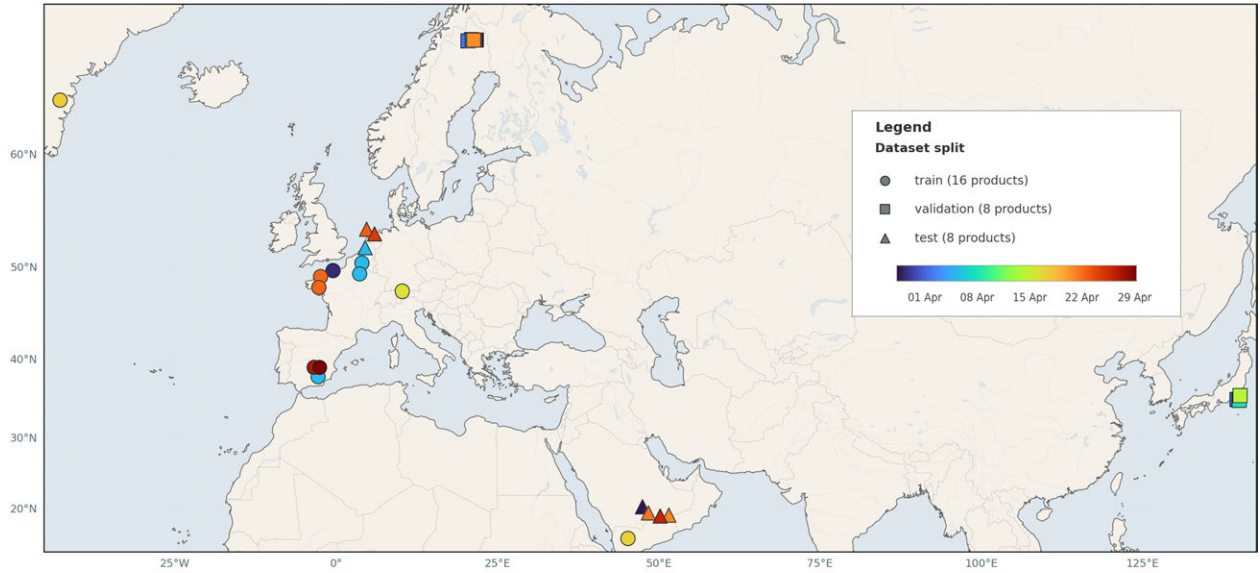


Figure 2 Geographical coverage and temporal distribution (April 2025) of Maya4 train/validation/test products.

sharding. Chunks define the access granularity for decoding and slicing, while sharding packs multiple chunks into fewer physical objects, reducing metadata operations and HTTP request overhead on object storage.

This layout is designed for learning workloads: chunk-wise reads support random patch sampling without loading full strips, and sharding improves throughput for multi-worker data loaders by amortizing network latency across neighboring chunks.

For completeness, we report the production storage settings used in this release. Each acquisition is processed as a full strip (about $60,000 \times 20,000$ samples) and stored at all released levels (raw, rc, rcmc, az). No release-time tiling, overlap, or padding is applied. Arrays are stored with 32×32 Zarr chunks, sharding enabled, and downstream user tiles cropped after loading.

Each Maya4 product follows the directory layout `product_name.zarr/`, with four top-level processing groups raw/, rc/, rcmc/, and az/; within each processing group, the same three subgroups are present—ancillary/ for auxiliary metadata, ephemeris/ for orbit/attitude information, and echoes/ for complex SAR arrays—which ensures a uniform hierarchy across all processing levels and simplifies programmatic traversal of the signal chain.

4 Potential Use Cases

4.1 End-to-End Compression

Numerous works are studying neural compression for on orbit data reduction [4, 19–22]. The inclusion of early-stage representations within the Maya4 dataset enables the development of data-driven models for *end-to-end SAR signal compression*. Let $s_{\text{raw}}[n, m]$ denote the sampled raw complex echo at range index n and azimuth index m . For compact operator notation, the stacked array is denoted $\mathbf{s}_{\text{raw}} \in \mathbb{C}^{N \times M}$. A learned compression model

$$f_{\theta} : \mathbf{s}_{\text{raw}} \rightarrow \mathbf{z}, \quad \mathbf{z} \in \mathbb{R}^d, \quad d \ll MN \quad (8)$$

can be trained to map the raw signal to a compact latent representation $\mathbf{z}$ using a neural encoder f_{θ} . The corresponding decoder g_{ϕ} reconstructs an approximation $\hat{\mathbf{s}}_{\text{raw}} = g_{\phi}(\mathbf{z})$. The objective minimizes reconstruction error

rate–distortion constraints:

$$\min_{\theta, \phi} \mathbb{E}[\|\mathbf{s}_{\text{raw}} - \hat{\mathbf{s}}_{\text{raw}}\|_F^2 + \lambda R(\mathbf{z})], \quad (9)$$

where $R(\mathbf{z})$ measures entropy or bit-rate, and λ controls the compression–quality trade-off.

When deployed onboard satellites, such a model could reduce telemetry volume prior to conventional focusing, transferring only $\mathbf{z}$ to the ground segment. In this setting, part of the processing burden shifts toward the onboard encoder, while the achievable downlink reduction depends on the selected bitrate and the acceptable reconstruction error.

4.2 Learned Focusing and Autofocusing

Maya4 provides paired intermediate levels that define a supervised mapping between raw signals and focused SAR imagery. Traditional SAR focusing is achieved through matched filtering and azimuth compression:

$$S_{\text{rcmc}}[n, \ell] = \mathcal{F}_m(s_{\text{rcmc}}[n, m]), \quad (10)$$

$$s_{\text{az}}[n, m] = \mathcal{F}_m^{-1}[S_{\text{rcmc}}[n, \ell] \cdot H_{\text{az}}[\ell]], \quad (11)$$

where $\mathcal{F}_m$ denotes the azimuth Fourier transform and $H_{\text{az}}[\ell]$ the focusing transfer function.

Beyond deterministic focusing, *learned autofocusing* extends this paradigm to estimate and correct residual phase errors directly from data. Traditional autofocusing techniques either directly estimate the residual phase error—exemplified by the Phase Gradient Autofocus (PGA) algorithm [23]—or iteratively optimize a phase-correction function $\phi[\ell]$ to enhance image sharpness or minimize image entropy [24]. In a learned setting, current works [25–27] estimate the residual phase term by a dedicated network and apply it multiplicatively in the azimuth-frequency domain:

$$\hat{S}_{\text{az}}[n, \ell] = S_{\text{rcmc}}[n, \ell] H_{\text{az}}[\ell] e^{-i\hat{\phi}[\ell]}, \quad (12)$$

$$\hat{s}_{\text{az}}[n, m] = \mathcal{F}_m^{-1}\{\hat{S}_{\text{az}}[n, \ell]\}, \quad (13)$$

$$\hat{\phi}[\ell] = \Phi_{\psi}(s_{\text{rcmc}}[n, m]). \quad (14)$$

where Φ_{ψ} is a neural phase-estimation operator parameterized by ψ . This formulation ensures that residual phase errors are correctly modeled as multiplicative distortions in the ℓ —consistent with the physics of coherent

4.3 Representation Learning

As modern literature highlights the potential of self-supervised learning for SAR [28–31], the hierarchical structure of Maya4 enables *representation learning*, wherein latent features are extracted without explicit manual labels. Given multiple intermediate processing levels

$$\{s_k[n, m]\}_{k=1}^K, \quad s_k \in \{s_{\text{raw}}, s_{\text{rc}}, s_{\text{rcmc}}, s_{\text{az}}\},$$

an encoder network E_θ maps each level to a corresponding feature embedding

$$\mathbf{h}_k = E_\theta(s_k), \quad (15)$$

with the goal of preserving *cross-level consistency*—that is, ensuring embeddings derived from physically related signal states remain close in the latent space while distinct acquisitions remain separable.

A standard contrastive objective, such as the InfoNCE loss, can be adopted to enforce this property:

$$\mathcal{L}_{\text{NCE}} = - \sum_i \log \frac{\exp(\text{sim}(\mathbf{h}_i^{(a)}, \mathbf{h}_i^{(b)})/\tau)}{\sum_j \exp(\text{sim}(\mathbf{h}_i^{(a)}, \mathbf{h}_j^{(b)})/\tau)}, \quad (16)$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity metric (typically cosine similarity), τ is a temperature hyperparameter controlling distribution sharpness, and the indices (a, b) denote paired representations of the same acquisition observed at different processing levels.

The learned representations serve as a compact, physically informed latent space that can be transferred to downstream analysis tasks.

4.4 Algorithm Benchmarking

Maya4 also serves as a comprehensive testbed for benchmarking classical and learned SAR algorithms. To ensure a consistent comparison, both the learned focusing network and a conventional processor are applied to the same motion-compensated input:

$$\hat{s}_{\text{az}}[n, m] = \mathcal{N}_\theta(s_{\text{rcmc}}[n, m]), \quad (17)$$

$$s_{\text{az}}^{(P)}[n, m] = \mathcal{P}(s_{\text{rcmc}}[n, m]). \quad (18)$$

The conventional output $s_{\text{az}}^{(P)}[n, m]$ is taken as the reference $s_{\text{az}}^*[n, m]$, and quantitative metrics are computed between $\hat{s}_{\text{az}}[n, m]$ and $s_{\text{az}}^*[n, m]$:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\|s_{\text{az}}^*\|_2^2}{\|s_{\text{az}}^* - \hat{s}_{\text{az}}\|_2^2} \right), \quad (19)$$

$$\text{SSIM} = \text{SSIM}(s_{\text{az}}^*, \hat{s}_{\text{az}}), \quad (20)$$

$$\gamma = \frac{|\sum_{n,m} s_{\text{az}}^*[n, m] \hat{s}_{\text{az}}[n, m]|}{\sqrt{\sum_{n,m} |s_{\text{az}}^*[n, m]|^2 \sum_{n,m} |\hat{s}_{\text{az}}[n, m]|^2}}. \quad (21)$$

Here, PSNR and SSIM quantify radiometric fidelity and structural similarity, respectively, while γ denotes the *complex coherence*, measuring phase and amplitude consistency between the reference and the reconstructed signals. With vectorized outputs $\mathbf{v}^* = \text{vec}(s_{\text{az}}^*[n, m])$ and $\hat{\mathbf{v}} = \text{vec}(\hat{s}_{\text{az}}[n, m])$, the corresponding cross-correlation matrix can be defined as

$$\mathbf{C} = \mathbb{E} \left[\begin{pmatrix} \mathbf{v}^* \\ \hat{\mathbf{v}} \end{pmatrix} \begin{pmatrix} \mathbf{v}^* & \hat{\mathbf{v}} \end{pmatrix}^* \right] = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

from which the empirical coherence estimate follows as

$$\gamma = \frac{|C_{12}|}{\sqrt{C_{11} C_{22}}}. \quad (23)$$

This unified framework evaluates amplitude accuracy, structural integrity, and complex-domain coherence between learned and conventional focusing outputs. Beyond direct quality assessment, the Maya4 benchmark supports systematic analyses of compression artefacts, robustness to noise and motion perturbations, and generalization across acquisition geometries and polarization modes—thus fostering reproducible and quantitative progress in radar-based machine learning.

5 Conclusions

This paper introduced Maya4, a cloud-native SAR dataset that exposes the full Sentinel-1 processing chain from raw echoes to focused imagery. By releasing intermediate signal representations together with physically grounded metadata, Maya4 bridges a longstanding gap between model-based radar processing and modern data-driven learning.

The dataset supports a broad class of machine-learning workflows that were previously difficult to study at scale with public data. In particular, the multi-level structure enables end-to-end signal compression, learned focusing and auto-focusing, self-supervised representation learning across processing states, and rigorous benchmarking against conventional processors in both image and complex domains.

Beyond a single benchmark, Maya4 provides a reproducible foundation for research at the intersection of radar physics and machine learning. We expect it to accelerate the development of physically informed algorithms, improve comparability across methods, and support a new generation of scalable SAR learning systems for Earth observation applications.

References

- [1] I. G. Cumming and F. H.-C. Wong, *Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation*. Boston: Artech House, 2005.
- [2] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, “A tutorial on synthetic aperture radar,” *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 1, pp. 6–43, 2013.
- [3] K. Kamirul, O. Pappas, and A. Achim, “R-Sparse R-CNN: SAR ship detection based on background-aware sparse learnable proposals,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.18959>
- [4] R. M. Asiyabi, A. Anghel, P. Rizzoli, M. Martone, and M. Datcu, “Complex-valued autoencoder for multi-polarization SLC SAR data compression with side information,” in *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023, pp. 1787–1790.
- [5] H. Wang, Y. Huang, L. Wu, Z. Guo, and N. Li, “Land cover classification of SAR images based on ascending and descending orbit feature optimization and data fusion,” *Remote Sensing Letters*, vol. 16, no. 8, pp. 880–889, 2025. [Online]. Available: <https://doi.org/10.1080/2150704X.2025.2513547>
- [6] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, “SENTINEL-1/2 – a curated dataset of georeferenced multi-sentinel-1/2 imagery for deep learning and

- data fusion,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W7, pp. 153–160, 2019.
- [7] C. Robinson, K. Malkin, N. Jojic, H. Chen, R. Qin, C. Xiao, M. Schmitt, P. Ghamisi, R. Hänsch, and N. Yokoya, “Global land-cover mapping with weak supervision: Outcome of the 2020 IEEE GRSS data fusion contest,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3185–3199, 2021.
 - [8] C. Rambour, N. Audebert, E. Koeniguer, B. Le Saux, M. Crucianu, and M. Datcu, “Flood detection in time series of optical and SAR images,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2020, 2020, pp. 1343–1349.
 - [9] L. Huang, B. Liu, B. Li, W. Guo, W. Yu, Z. Zhang, and W. Yu, “OpenSARShip: A dataset dedicated to sentinel-1 ship interpretation,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 195–208, 2018.
 - [10] D. Bonafilia, B. Tellman, T. Anderson, and E. Issenberg, “Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 835–845.
 - [11] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi, “HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation,” *IEEE Access*, vol. 8, pp. 120 234–120 254, 2020.
 - [12] J. Li, C. Qu, and J. Shao, “Ship detection in SAR images based on an improved faster R-CNN,” *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA)*, pp. 1–6, 2017.
 - [13] X. Sun, X. Wang, C. Wang, W. Diao, Y. Zhang, K. Fu, and Z. Yan, “SARDet-100K: Towards open-source benchmark and toolkit for large-scale SAR object detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.
 - [14] J. C. Curlander and R. N. McDonough, *Synthetic Aperture Radar: Systems and Signal Processing*. New York: Wiley-Interscience, 1991.
 - [15] R. Torres, P. Snoeijs, D. Geudtner, D. Bibby, M. Davidson, E. Attema, P. Potin, B. Rommen, N. Floury, M. Brown, I. N. Traver, P. Deghaye, B. Duesmann, B. Rosich, N. Miranda, C. Bruno, M. L’Abbate, R. Croci, A. Pietropaolo, M. Huchler, and F. Rostan, “Gmes sentinel-1 mission,” *Remote Sensing of Environment*, vol. 120, pp. 9–24, 2012.
 - [16] *Sentinel-1 Level-1 Detailed Algorithm Definition*, European Space Agency, 2022, issue 2.5. [Online]. Available: <https://sentiwiki.copernicus.eu/web/document-library>
 - [17] S. N. Madsen, “Estimating the doppler centroid of sar data,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 134–140, 1989.
 - [18] ESA-PhiLab, “sarpyx,” GitHub repository, 2026, accessed 7 Apr. 2026. [Online]. Available: <https://github.com/ESA-PhiLab/sarpyx>
 - [19] Sujit Bhattacharya, S. Bhattacharya, Thomas Blumensath, T. Blumensath, Bernard Mulgrew, B. Mulgrew, Mike E. Davies, and M. Davies, “Fast Encoding of Synthetic Aperture Radar Raw Data for Compressed Sensing,” *IEEE/SP 14th Workshop on Statistical Signal Processing*, pp. 448–452, Aug. 2007, mAG ID: 2107664115.
 - [20] M. Martone, M. Villano, M. Younis, and G. Krieger, “Efficient onboard quantization for multichannel SAR systems,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1859–1863, 2019.
 - [21] N. Gollin, M. Martone, E. Imbembò, M. Ghiglione, S. Knoll, G. Krieger, and P. Rizzoli, “AI-BAQ: Deep learning for adaptive SAR raw data quantization,” *IEEE Transactions on Geoscience and Remote Sensing*, 2025.
 - [22] G. Pilikos, M. Azcueta, R. Camarero, and N. Floury, “Raw SAR data compression with deep learning,” in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, 2024, pp. 2546–2549.
 - [23] D. E. Wahl, P. Eichel, D. Ghiglia, and C. Jakowatz, “Phase gradient autofocus—a robust tool for high resolution sar phase correction,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 827–835, 2002.
 - [24] J. Wang and X. Liu, “SAR minimum-entropy autofocus using an adaptive-order polynomial model,” *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 512–516, 2006.
 - [25] R. Liu, L. Mi, and Z. Chen, “AFNet: Adaptive fusion network for remote sensing image semantic segmentation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7871–7886, 2020.
 - [26] W. Pu, “SAE-Net: A deep neural network for SAR autofocus,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
 - [27] Z. Liu, S. Yang, Z. Feng, Q. Gao, and M. Wang, “Fast SAR autofocus based on ensemble convolutional extreme learning machine,” *Remote Sensing*, vol. 13, no. 14, p. 2683, 2021.
 - [28] E. Dalsasso, L. Denis, and F. Tupin, “As if by magic: Self-supervised training of deep despeckling networks with MERLIN,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, p. 1–13, 2022.
 - [29] M. Muzeau, J. Frontera-Pons, C. Ren, and J. Ovarlez, “SAFE: A SAR feature extractor based on self-supervised learning and masked siamese vision transformers,” *arXiv preprint arXiv:2407.00851*, 2024.
 - [30] J. Amao-Oliva, N. Foix-Colonier, and F. Sica, “Joint compression and despeckling by SAR representation learning,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 220, p. 524–534, 2025.
 - [31] M. Wang, H. Bi, Y. Feng, L. Xin, S. Gong, T. Wang, Z. Yan, P. Wang, W. Diao, and X. Sun, “A complex-valued SAR foundation model based on physically inspired representation learning,” *arXiv preprint arXiv:2504.11999*, 2025.