SLC SAR Preprocessing — Complete Project Document

Purpose: This document defines an end-to-end, production-grade preprocessing pipeline for a single SLC (Single-Look Complex) SAR image intended to enable the highest-performance downstream maritime tasks: vessel detection, classification, dimension estimation, and direction (radial velocity) prediction. It assumes resources are available and documents tools, commands, parameters, outputs, quality checks, and advanced algorithmic extensions (DL/ML/hybrid physics-based methods).

Table of Contents

---

1. Mission & Goals

Create physics-preserved, calibrated, denoised, and geocoded SLC-derived products that maximize signal-to-noise for small/low-RCS vessel detection while preserving phase information for coherence/ATI-based motion cues. Deliver high-quality feature stacks for detectors and complete provenance for legal/audit needs.

---

2. Resources & Infrastructure (recommended)

- **Data:** SLC (complex float32) + orbit precise ephemerides (POD/POE) + DEM (SRTM/ArcticDEM) + optional revisits (multi-temporal) + optional polarimetric channels.
- **Environment / Software:**
    - ESA SNAP (S1TBX) with snappy for Python integration
    - ISCE or GAMMA for interferometry / phase-critical ops
    - GDAL / Rasterio, NumPy, SciPy
    - rsgislib, Orfeo ToolBox (OTB)
    - PyTorch (GPU), TensorFlow optional
    - SAR2SAR / Speckle2Void repositories for DL denoising
    - Docker for reproducible dev environments
- **Ancillary data:** wind/wave models, tide models, AIS logs, operator-provided ground truth (if available)

---

3. Core Preprocessing Principles

- **Preserve precision & phase.** Keep complex SLC float32 for any phase-dependent processing (coherence, ATI). Never downcast to 8-bit for algorithmic work.
- **Apply physics-first corrections.** Calibration and orbit correction before statistical methods.
- **Denoise conservatively then refine.** Two-stage denoising: non-local/classical first, DL second.
- **Avoid irreversible operations before archiving.** Maintain raw calibrated complex copy for traceability.
- **Do speckle filtering in slant-range where possible.** Geocoding/resampling changes statistics.

---

4. Detailed Step-by-Step Preprocessing Pipeline

Step 0 — Metadata & Sanity Checks

- Verify SLC integrity, sensor mode (TOPS/IW/SM), polarization channels, orbit files, PRF, pixel spacing, burst structure.
- Confirm acquisition geometry (incidence angles grid), noise LUT availability.

Deliverable: metadata.json (sensor, start/end time, polarizations, product type, pixel spacing, orbit ID)

Step 1 — Apply precise orbits & time corrections

- Use precise orbit (POD/POE) to improve geolocation and enable accurate interferometry.
- Tool: SNAP 'Apply Orbit File' or ISCE/GAMMA equivalent.

Why: accurate geolocation and phase stability are required for coherence/ATI.

Step 2 — TOPS split / Deburst (if applicable)

- For TOPS-mode acquisitions (e.g., Sentinel-1 IW), split bursts, align burst overlap regions, and remove burst boundaries carefully.
- Keep complex SLC format throughout.

Tool: SNAP TOPSAR-Split / Deburst operations.

Step 3 — Phase-preserving Radiometric Calibration

- Convert SLC counts to calibrated backscatter (sigma0 or gamma0). Prefer multiplicative operations applied to complex samples to preserve phase.
- Verify algorithm preserves complex samples.

Tool: SNAP Calibration (check 'output as amplitude, sigma0' vs 'complex-preserving' behavior). ISCE/GAMMA if strict phase control needed.

Step 4 — Thermal-noise handling (phase-aware)

- If phase/ATI will be used: perform thermal noise correction in complex domain or postpone until phase calculations are complete. Avoid SNAP intensity-only noise operators that may alter phase unexpectedly.
- If phase is not needed: intensity-only thermal-noise correction is acceptable.

Why: thermal noise biases intensity and impacts small-target detection; but some tools handle it in intensity-only ways that break complex-phase information.

Step 5 — AOI Subset / Tiling

- Crop to area-of-interest to improve processing speed and GPU memory usage. Use overlapping tiles (10–20% overlap) for DL tiling safety.

Deliverable: tiled SLCs in complex float32.

Step 6 — Precise Co-registration (multi-temporal only)

- For multi-temporal stacks: co-register SLCs to sub-pixel accuracy using complex cross-correlation. Required for: coherence, multi-temporal denoising, speckle2self strategies.
- Tools: ISCE co-registration routines or SNAP + custom complex-domain registration.

Step 7 — Two-stage Speckle Reduction (the heart of preprocessing)

Stage A — Conservative classical filter (non-local / Refined-Lee):

- Purpose: reduce variance while preserving point scatterers. Use small-to-medium windows to prevent blurring of tiny ships.
- Candidate filters: Refined-Lee, NL-SAR (non-local), SAR-BM3D.
- Suggested starting params (sensor dependent):

- Refined-Lee window: 7×7
- NL-SAR patch: 7×7, search window: 21×21

Stage B — SAR-aware deep denoiser (fine cleanup):

- Use self / semi-supervised models: SAR2SAR, Speckle2Void, Noise2Noise variants adapted to multiplicative noise with log-domain training.
- Training sources: multi-temporal stacks (noisy pairs), simulated clean-from-forward-model data, any available labeled clean patches.

Why two-stage: classical filters remove bulk speckle without hallucination; DL stage learns and cleans residuals while preserving scattering centers.

Step 8 — Optional controlled multilooking

- If SNR is extremely poor, apply minimal multi-look (e.g., 2× in range) but avoid heavy multi-looking when small vessel size estimation is required.

Note: multi-looking reduces speckle at cost of resolution — choose only if necessary.

Step 9 — Preserve complex copy for phase-based cues

- Keep a copy of the calibrated complex SLC (pre-denoising or minimally denoised in complex-preserving way) for coherence/ATI and velocity estimation. This is the canonical archive for forensic work.

Deliverable: SLC_calibrated_complex_v1 (archive)

Step 10 — Compute ATI / Coherence maps (if temporally adjacent SLCs exist)

- Compute complex cross-correlation between temporally close SLCs to produce coherence maps.
- Compute ATI interferograms / radial velocity estimates if along-track baselines or dual-antenna data available.

Tools: ISCE, GAMMA; SNAP with phase-aware operators (if validated).

Step 11 — Build detection feature stack (multi-channel)

Produce a multi-band stack (per tile) containing:

1. denoised log-amplitude (primary detection input)
2. original calibrated log-amplitude (for comparisons)
3. coherence map (if available)
4. ATI magnitude/phase (if available)
5. polarimetric decompositions (Pauli, Cloude-Pottier) if pol data present
6. incidence-angle band and look-direction band
7. texture features: local mean, local variance, wavelet bands, GLCM-based local contrast

Format: float32 GeoTIFF multi-band or HDF5/Numpy .npz stacks for DL.

Step 12 — Terrain correction / precise geocoding

- Geocode the denoised intensity and feature stack to map coordinates using a DEM and the precise orbit.
- Perform speckle filtering *before* geocoding where possible (in slant-range) to preserve speckle statistics.
- Use high-precision resampling (e.g., cubic convolution) but be aware of resampling effects on small targets.

Deliverable: geocoded tile datasets with retained float precision and per-pixel incidence angle.

Step 13 — QC, metrics & provenance

- Generate QC metrics: ENL (Equivalent Number of Looks) pre/post, local SNR improvement, edge preservation score (e.g., SSIM vs minimally denoised copy), phase residual histograms, tiled visual QA images.

- Save provenance: chain of operations, parameters, timestamps, software versions, and operator notes in provenance.json.

Deliverable: QC_report.pdf, provenance.json, and all intermediate product copies.

Deep Learning & Advanced Algorithms for SAR Preprocessing

(*for small/low-RCS vessel detection, classification, dimension estimation, direction prediction*)

1. Self-Supervised / Unsupervised Denoising

These methods learn to reduce speckle **without clean ground truth**, which is critical in SAR.

SAR2SAR

- **Paper**: Dalsasso et al., IEEE TGRS 2021
- **Idea**: Train on pairs of SAR images from the *same scene* with independent noise realizations (multi-temporal or split-look).
- **Why good**: Keeps fine vessel scatterers intact while lowering variance.
- **Implementation**: Residual U-Net, log-domain training, gamma-likelihood loss.
- **Tip**: Ensure multi-temporal registration < 0.5 px.

Speckle2Void

- **Idea**: Adaptation of Noise2Void — mask random pixels and predict from surroundings.
- **Why good**: Needs only single noisy images, no clean target.

- **Implementation**: Blind-spot CNN with masked convolutions.
- **Tip**: Combine with augmentations to prevent overfitting to texture.

Noise2Noise (SAR variant)

- **Idea**: Train with two noisy versions of same scene — network learns clean mapping implicitly.
- **Note**: Works best if temporal decorrelation is minimal.

---

2. Complex-Domain / Physics-Aware Networks

Go beyond magnitude-only and handle **real + imaginary** SAR data directly.

Complex-Valued CNNs

- **Approach**: Use complex convolutions, activations, and batch norms.
- **Benefit**: Preserve phase information for downstream interferometry/ATI.
- **Tools**: `complexPyTorch`, custom layers.
- **Tip**: Useful if you later do motion detection or coherence-based vessel velocity.

Magnitude-Phase Dual-Branch Networks

- **Approach**: Split complex SLC into magnitude & phase channels → process in parallel → fuse.
- **Benefit**: Lets magnitude branch handle amplitude denoising and phase branch handle coherence stability.

Hybrid Physics-DL Models

- Embed SAR imaging equations (e.g., speckle model, system PSF) into loss or network layers.
- Example: Learnable despeckle filter with multiplicative noise prior in network.

---

3. Hybrid Classical + DL Filtering

Use **physics-based filtering first**, then DL for refinement.

Workflow Example:

1. Refined-Lee →
2. DL residual denoiser (U-Net/ResNet) →
3. Edge-preserving loss to avoid blurring vessels.

Why good:

- Classical stage ensures physics preservation.
- DL removes residual speckle without damaging point scatterers.

---

4. Domain Adaptation & Transfer Learning

Bridge gap between **optical DL pretraining** and SAR.

Domain-Adversarial Training (DANN)

- Train with adversarial loss so SAR features match optical pretraining features.

Adaptive Batch Normalization (AdaBN)

- Re-estimate BN statistics on SAR domain before fine-tuning.

Style Transfer Preprocessing

- Convert SAR texture stats toward source domain model was trained on — caution: may distort physics.

---

5. Synthetic Data Simulation & Augmentation

Boost training data diversity.

Simulation

- **EM scattering simulation** for ships (e.g., Xpatch, FEKO).
- Simulate wakes with hydrodynamic models.
- Inject into real sea clutter from actual SAR.

Augmentation

- Rotation, flip, scale (preserving aspect ratio).
- Incidence angle variation (gamma correction in log-domain).
- Noise injection: controlled speckle variance changes.
- Weather simulation: wind speed maps → adjust clutter texture.
- Occlusion: partial wakes or partial vessel returns.

---

6. Super-Resolution Enhancement

Increase apparent resolution while preserving physics.

ESRGAN-SAR

- Adapt ESRGAN with speckle-aware loss.

- Train on high-res TerraSAR-X as target, low-res Sentinel-1 as input.

Deep Back Projection Networks

- Reconstruct higher-res images using SAR forward model.

---

7. Multi-Task Preprocessing Networks

Train one network to **denoise, super-resolve, and normalize incidence angle** in a single pass.

- **Architecture**: U-Net with multiple output heads.
- **Loss**: Multi-task weighted sum (denoising loss + SR loss + angle normalization loss).
- **Benefit**: One inference pass gives all enhanced products.

---

8. Loss Functions for SAR Preprocessing

Specialized losses to match SAR noise/statistics.

- **Gamma-Likelihood Loss** (matches speckle statistics).
- **Log-Domain MSE** (stabilizes multiplicative noise).
- **Perceptual Loss** (from VGG/ResNet trained on SAR patches).
- **Edge Loss** (Sobel/Laplacian edge map MSE).
- **Physics Regularizer**: penalize outputs that violate known SAR scattering models.