Towards end-to-end SAR raw data handling: RCMC data compression using hyperprior autoencoders

Cédric Léonard[a,b], Roberto Del Prete[c], and Francescopaolo Sica[d]
[a]Technical University of Munich, Munich, Germany
[b]Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany
[c]ESA Φ-lab, European Space Research Institute (ESRIN), Frascati, Italy
[d]Department Aerospace Engineering, University of the Bundeswehr Munich, Neubiberg, Germany

Abstract

Synthetic Aperture Radar (SAR) is a fundamental imaging technique in Remote Sensing. While modern SAR systems offer unprecedented resolution and coverage, they generate large volumes of raw data that cannot be processed directly onboard and must be transmitted to the ground station, straining downlink bandwidth. In an effort to bring efficient SAR data compression onboard, this paper explores the possibility of compressing intermediate representations of SAR data. In particular, we experiment with the compression of Range Cell Migration Corrected (RCMC) data using hyperprior autoencoders. To this end, we adapt existing deep learning compression models to the specific characteristics of SAR RCMC data and introduce a task-aware training strategy that enforces coherence preservation. Experimental results show that hyperprior models achieve up to 9 dB PSNR improvement over traditional codecs, while preserving complex coherence up to 0.95 at high bitrates. The code is released at `https://github.com/CedricLeon/MAYA_DC`.

1 Introduction

Synthetic Aperture Radar (SAR) is a crucial sensing modality for Earth Observation. With the advent of new SAR missions and advanced sensor technologies, the amount of data generated is expected to increase dramatically. Next-generation missions, such as Sentinel-1 NG [8], promise improved spatial resolution, shorter revisit times, and wider swaths. The improvements come at the cost of a substantial increase in data volume, which poses significant challenges for data handling, as onboard storage and downlink bandwidth remain limited resources. Data compression thus becomes a critical component of the SAR processing chain, and should ideally be applied as early as possible in the acquisition pipeline to minimize transmission costs. Traditional onboard compression schemes, such as Block Adaptive Quantization (BAQ) [11] or its acquisition-dependent extension, Flexible Dynamic BAQ (FDBAQ) [3], employed on Sentinel-1, offer efficient yet handcrafted solutions.

In parallel, the field of Learned Image Compression (LIC) [4] has rapidly evolved in computer vision, where hyperprior autoencoder architectures [5, 13] have established a new state of the art. These models achieve remarkable Rate–Distortion (RD) trade-offs by jointly optimizing compact latent representations and entropy models for efficient coding [4]. Despite their success in on-ground optical imaging, their potential for SAR data remains largely unexplored, especially onboard.

This work takes a step toward efficient onboard SAR data compression by investigating the use of autoencoders to compress intermediate representations of the SAR image formation chain. Specifically, we experiment with Range Cell Migration Corrected (RCMC) data, i.e., raw data that has been compressed along the range direction and corrected for platform-induced migration effects. RCMC data offers a good balance between computational cost and information richness. It presents higher structural fidelity than raw data while being one computationally expensive processing step away, namely azimuth compression, from Single-Look Complex (SLC) images.

Our contributions are threefold:

- we propose to perform compression at the RCMC stage, offering a trade-off between information content and computational cost;
- we introduce a task-aware distortion function that explicitly enforces coherence preservation;
- we evaluate compression performance through azimuth focusing, enabling assessment of the final SLC image quality.

2 Related work

Compressing SAR raw data has been an active area of research to manage the increasing data volumes from modern SAR systems. The current operational standard for Sentinel-1 is the FDBAQ algorithm [3], which adaptively allocates quantization bits based on local Signal-to-Noise Ratio (SNR) estimates, achieving variable bit rates while maintaining radiometric quality across heterogeneous scenes. FDBAQ extends traditional BAQ [11] by dynamically selecting from multiple quantizers depending

on the backscatter intensity, resulting in improved performance over earlier entropy-constrained methods, particularly for bright targets such as urban areas.

Several recent works explore alternatives to FDBAQ. Asiyabi et al. [2] evaluate JPEG2000 [14] for SAR raw data compression, demonstrating the superior performance of JPEG2000 compared to BAQ. To improve BAQ in an acquisition-dependent compression method, Gollin et al. [9] introduced AI-BAQ, a deep learning method that predicts optimal bitrate maps directly from raw data. They train a lightweight Convolutional Neural Network (CNN) in a supervised manner to directly link raw data and the focused domain to satisfy some target performance metrics, such as SQNR or InSAR coherence. This data-driven strategy enhances the performance of BAQ without requiring onboard focusing or a priori scene information. As AI-BAQ predicts quantization maps, it essentially differs from our approach, where neural compressors learn end-to-end latent representations of pre-focused signals.

Further along the processing chain, LIC methods have been applied to focused SAR products. Asiyabi et al. [1] propose a complex-valued autoencoder architecture for compressing SLC data, leveraging dual-polarization channels as side information to improve reconstruction quality. These results indicate that learned representations can be effective for SAR data once geometric corrections and focusing have been applied, motivating the exploration of LIC methods at intermediate processing levels such as RCMC.

The feasibility of onboard SAR focusing is also advancing rapidly as Mandapati et al. [12] demonstrate real-time floating-point SAR focusing on FPGA hardware. This development suggests that computational constraints for spaceborne focusing are becoming less prohibitive, opening the possibility of generating focused products onboard and applying LIC methods directly in orbit to minimize downlink requirements.

3 Method

In this section, we detail our end-to-end compression pipeline for RCMC data. We first present the Maya4 dataset and its intermediate representations, then introduce LIC methods, and finally describe the modifications required to adapt them to RCMC data.

3.1 Maya4 Dataset

The Maya4 dataset[1] provides multi-level intermediate signal representations from Sentinel-1 Stripmap acquisitions. Unlike conventional datasets, which provide only Level-1 (SLC) products, Maya4 includes raw baseband echoes (`raw`), range-compressed data (`rc`), range cell migration-corrected data (`rcmc`), and fully focused SLC images (`az`). These representations correspond to physically and mathematically distinct stages of the SAR focusing pipeline, which enable targeted machine learning at each level.

[1] `https://huggingface.co/Maya4`

Given the complex raw echo signal $\mathbf{s}_{\text{raw}}(t, \eta)$, where t denotes the fast-time (range) and η the slow-time (azimuth), the processing chain [15] applies three transformations:

- **Range compression:** Matched filtering in the fast-time dimension,

$$\mathbf{s}_{\text{rc}}(t, \eta) = \mathbf{s}_{\text{raw}}(t, \eta) * h_r(t) \tag{1}$$

  where $h_r(t)$ is the time-reversed transmitted chirp.

- **Range Cell Migration Correction (RCMC):** Compensation for target migration caused by platform motion $\Delta R(\eta)$,

$$\mathbf{s}_{\text{rcmc}}(t, \eta) = \mathbf{s}_{\text{rc}}(t + \Delta R(\eta)/c,\, \eta)\,, \tag{2}$$

  where c is the speed of light.

- **Azimuth compression:** Focusing the signal along the slow-time dimension using matched filtering,

$$\mathbf{s}_{\text{az}}(t, \eta) = \mathcal{FFT}_{\eta}^{-1}[\mathcal{FFT}_{\eta}(\mathbf{s}_{\text{rcmc}}(t, \eta))\, H_{\text{az}}^{*}(f_\eta)]\,, \tag{3}$$

  with the matched filter $H_{\text{az}}(f_\eta) = \exp\left[-j\frac{4\pi f_\eta^2}{2K_a c/f_c}\right]$, where K_{ac} is the Doppler chirp rate and f_c is the carrier frequency.

The Maya4 dataset is distributed in `zarr` format, supporting chunked and scalable access, which enables the development of deep learning models for a wide range of applications directly in the radar signal domain, such as compression, focusing, or representation learning. For example, in this paper, the RCMC data serves as input to compression algorithms, whose predictions are later evaluated against the corresponding SLCs, providing deeper insight into the consequences of compression.

3.2 Neural Compression

Learned Image Compression (LIC) uses neural networks to perform lossy data compression. At the intersection of deep learning and information theory, LIC has been shown to outperform traditional handcrafted compression algorithms such as JPEG or JPEG2000 [4]. Conventional LIC architectures resemble autoencoders [10] and are trained end-to-end to jointly minimize the rate $\mathcal{R}$, i.e., the size of

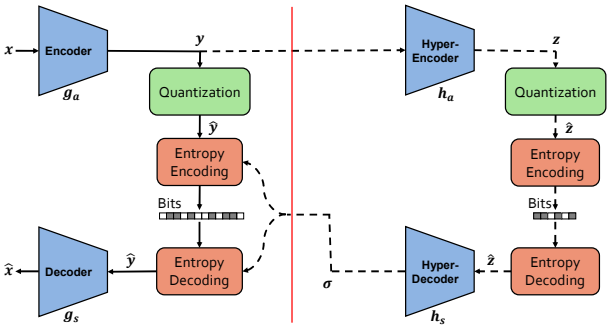

Figure 1 Architectures of the two neural compressors: FP uses only the left half; SH uses the complete diagram.

the compressed data, and the distortion $\mathcal{D}$ between the original and reconstructed image. In the loss $\mathcal{L}$, the tradeoff is determined by the Lagrange multiplier λ:

$$\mathcal{L} = \mathcal{R} + \lambda \cdot \mathcal{D} \tag{4}$$

The overall architecture of a conventional LIC model is illustrated in Figure 1. A first *encoder* network transforms the original image to a more compressible latent space, which is then quantized and losslessly encoded using standard entropy coding algorithms, to generate a compressed bitstream. During decompression, the generated bitstream is losslessly decoded to retrieve the quantized latents, which are then transformed back to pixels by the second *decoder* network. In LIC, the encoder g_a and the decoder g_s serve as parametric alternatives to the *analysis* and *synthesis* transforms used in traditional compression codecs, such as the wavelet transform in JPEG2000 [14].

Training these autoencoder architectures [5, 13] does not require any labels; instead, the model faces a RD optimization problem. Its goal is to minimize the expected length of the bitstream as well as the distortion of the reconstructed image with respect to the original. Therefore, Eq. (4) expands with:

$$\mathcal{R} \;=\; \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{y}}(\lfloor g_a(x)\rceil)\right] \tag{5}$$

$$\mathcal{D} \;=\; \mathbb{E}_{x\sim p_x}\left[d\big(x, g_s(\lfloor g_a(x)\rceil)\big)\right] \tag{6}$$

Where x is the input image and p_x its associated, unknown distribution, $\lfloor\cdot\rceil$ symbolizes the quantization operation, y corresponds to the latents, and $p_{\hat{y}}$ is the discrete entropy model, with $\hat{y} = \lfloor y \rceil$. The distortion $\mathcal{D}$ is computed using a distortion metric $d(\cdot)$ such as the Mean Squared Error (MSE) or Structural Similarity Index Measure (SSIM).

Entropy modeling requires a prior probability model. Standard practice uses a fully factorized distribution that assumes no statistical dependencies in the latent distribution. A Variational Auto-Encoder (VAE) architecture using a fully factorized prior model was introduced by Ballé et al. [4]. We name this model Factorized Prior (FP), its architecture is depicted on the left half of Figure 1.

To enhance compression performance, one can introduce a new set of variables z to capture the dependencies of the latent representations. This side information can then be used to estimate $\hat{\sigma}$, the distribution of the standard deviations of each latent variable, which improves our entropy prior model. In practice, this method requires two additional transformations implemented as neural networks: h_a, the *hyper-encoder*, and h_s, the *hyper-decoder*. In this split-paradigm, the side information z must also be quantized and compressed, demanding a second entropy model $p_{\hat{z}}$[2]. With this additional information to encode, the rate $\mathcal{R}$ of Eq. (5) becomes:

$$\begin{aligned}\mathcal{R} \;&= \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{y}}(\lfloor g_a(x)\rceil)\right] \\ &+ \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{z}}(\lfloor h_a(y)\rceil)\right]\end{aligned} \tag{7}$$

[2]As there are no prior beliefs about this distribution, a fully factorized density model is used again.

This new modeling of the latent distribution was introduced by Ballé et al. [5] and is illustrated on the right half of Figure 1. We refer to the complete architecture as Scale Hyperprior (SH), the main model used in this work[3].

3.3 RCMC Data Compression

Figure 2 depicts the complete project pipeline. Using the Maya4 dataset, we sample intermediate representations of partially focused Sentinel-1 images (`rcmc`) for compression. The images are then reconstructed as $\widehat{\texttt{rcmc}}$ and undergo azimuth compression. The obtained reconstructed SLCs $\widehat{\texttt{az}}$ are then compared to the regular SLCs `az`.

As the LIC compressors implemented in CompressAI [6] are designed for Computer Vision applications, a few modifications are necessary to adapt the FP and SH models to Maya4. First, the input channels are shrunk to two for the real and imaginary parts of the RCMC data. Then, we modify the loss function to account for the specific characteristics of SAR data and its applications.

Interferometric SAR (InSAR) relies on the phase correlation between acquisitions, which is directly determined by the interferometric coherence. As quantization and compression introduce decorrelation, which degrades coherence, any compression scheme intended for SAR products must explicitly preserve phase relationships to maintain downstream interferometric usability [7]. To ensure that our compression method maintains interferometric fidelity, we enrich the distortion component of the loss, Eq. (6), with two terms ensuring coherence conservation.

$$\begin{aligned}\mathcal{D} \;&= \|x - \hat{x}\|^2 \\ &+ \delta_{\text{kde}} \int \left|\, \hat{p}_{|\hat{x}|}(t) - \hat{p}_{|x|}(t) \,\right| \mathrm{d}t \\ &+ \delta_{\text{coh}} \left(1 - \frac{\left|\mathbb{E}[\hat{x}\cdot x^*]\right|}{\sqrt{\mathbb{E}[|\hat{x}|^2]\cdot\mathbb{E}[|x|^2]}}\right)\end{aligned} \tag{8}$$

Eq. (8) defines a composite distortion loss combining MSE, Kernel Density Estimation (KDE), and coherence terms, weighted by δ_{kde} and δ_{coh}, added to the loss function Eq. (4). $\hat{x}$ is the reconstructed image after compression and is equivalent to $g_s(\lfloor g_a(x)\rceil)\big)$ as entropy encoding and decoding are lossless processes, and $\hat{p}_{|\cdot|}$ is a Gaussian KDE.

4 Results

Due to the large size of RCMC tiles, they must be divided into smaller patches before being fed to the LIC compressor. As the signal has already been compressed in the fast-time dimension, the relevant information in the range direction is largely confined within a single resolution cell. However, this is not the case for the slow-time dimension (azimuth), where additional information is required for accurate azimuth compression. To estimate the number of azimuth cells required to capture most of a pixel's energy, we

[3]We implement the FP and SH models using the CompressAI framework [6] in which the models are respectively named `bmshj2018-factorized` and `bmshj2018-hyperprior`.

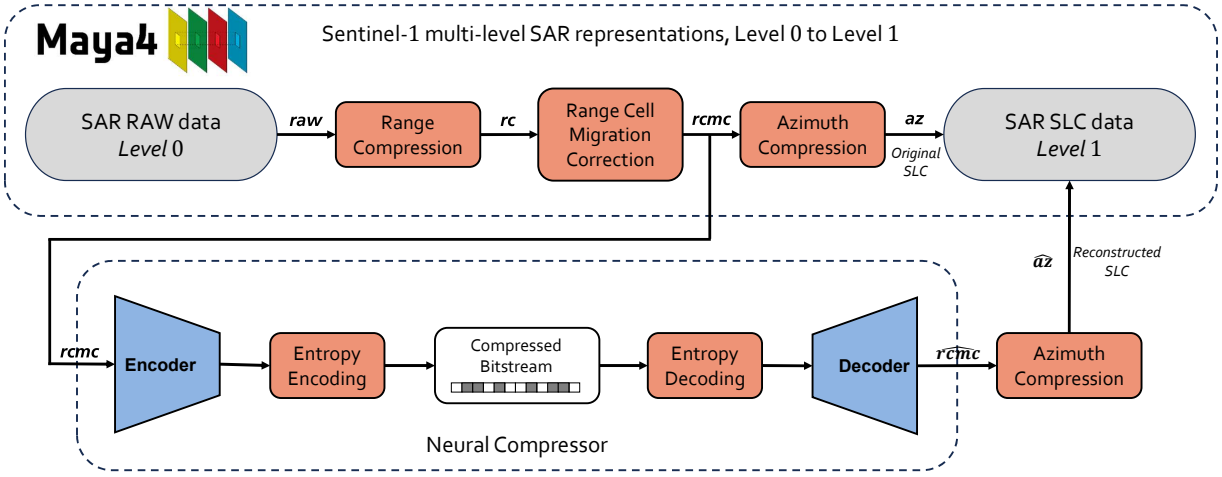

Figure 2 Method diagram and processing pipeline. Maya4 `rcmc` data is used as input for the neural compressor. The reconstructed $\widehat{\texttt{rcmc}}$ data then undergoes azimuth compression for comparison with the original SLC images `az`.

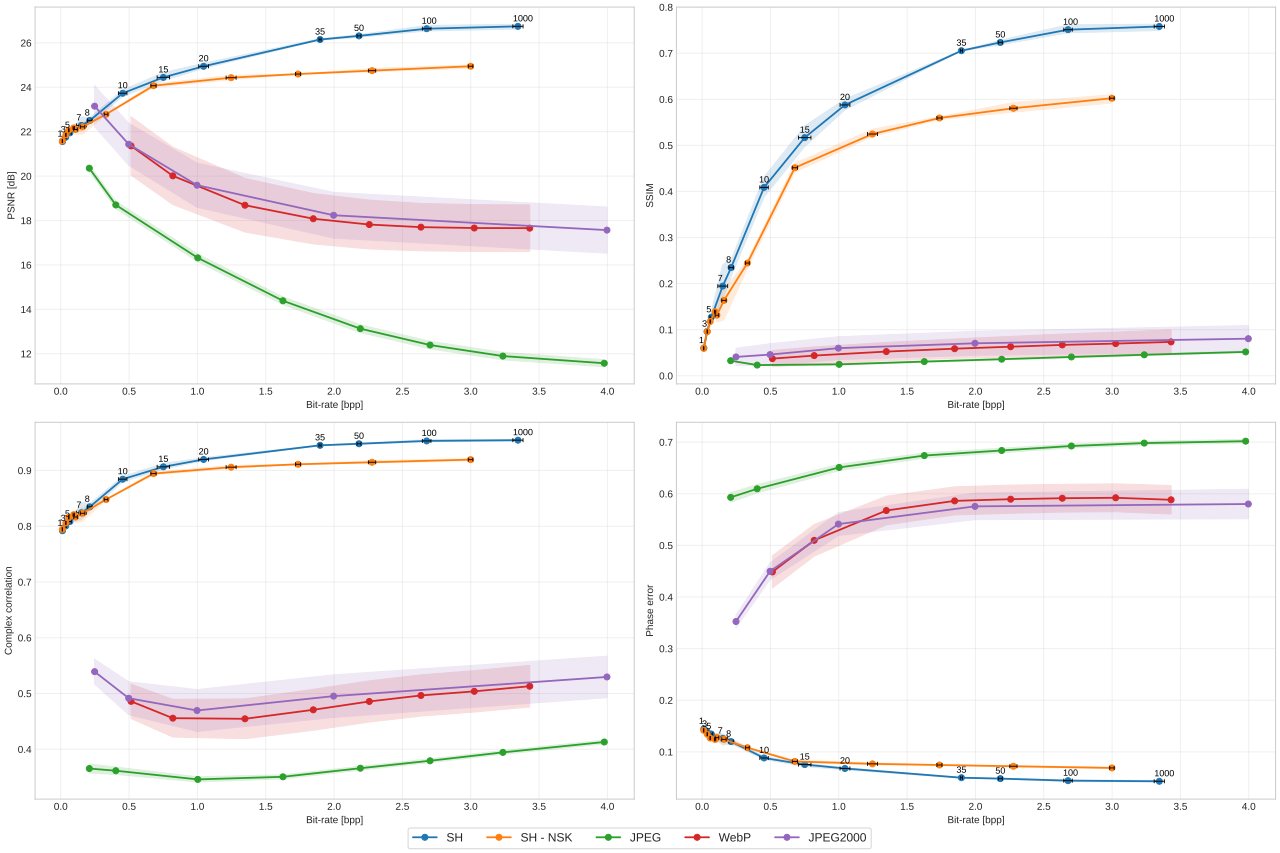

Figure 3 RD curves of the models Scale Hyperprior (SH), and SH-NSK using Non-Square Kernels, alongside three traditional codecs JPEG, WebP, and JPEG2000. Error bands correspond to the standard deviation of the quality metrics across seeds for the models and the standard error for the traditional codecs. Error bars show the standard deviation of the bpp across seeds and are not represented for traditional codecs, as the rate highly varies across patches.

start from the length of the synthetic aperture $L_{\text{SA}} = R_0 \cdot \Theta$, where Θ is the angular span and R_0 the range of closest approach. Then, we can approximate Θ as the ratio of the transmitted wavelength μ and the physical antenna length L_{ant}: $\Theta \approx \frac{\mu}{L_{\text{ant}}}$. Finally, with $N_{\text{az}} = \frac{L_{\text{SA}}}{\Delta x_{\text{az}}}$, where Δx_{az} is the ground-projected size, we obtain $N_{\text{az}} \approx \frac{R_0\,\mu}{L_{\text{ant}}\,\Delta x_{\text{az}}}$ resulting in an estimate of approximately 486 azimuth cells using Sentinel-1 characteristics. After rounding up to the nearest power of two, we used a buffer of 512 pixels on each side in the slow-time direction, leading to a final rectangular patch size of 1536×512. This buffer is dropped after azimuth compression, so that loss and other metrics are computed only on the 512×512 central patch, which reduces training efficiency by computing the loss on only one third of the reconstructed region, but avoids boundary

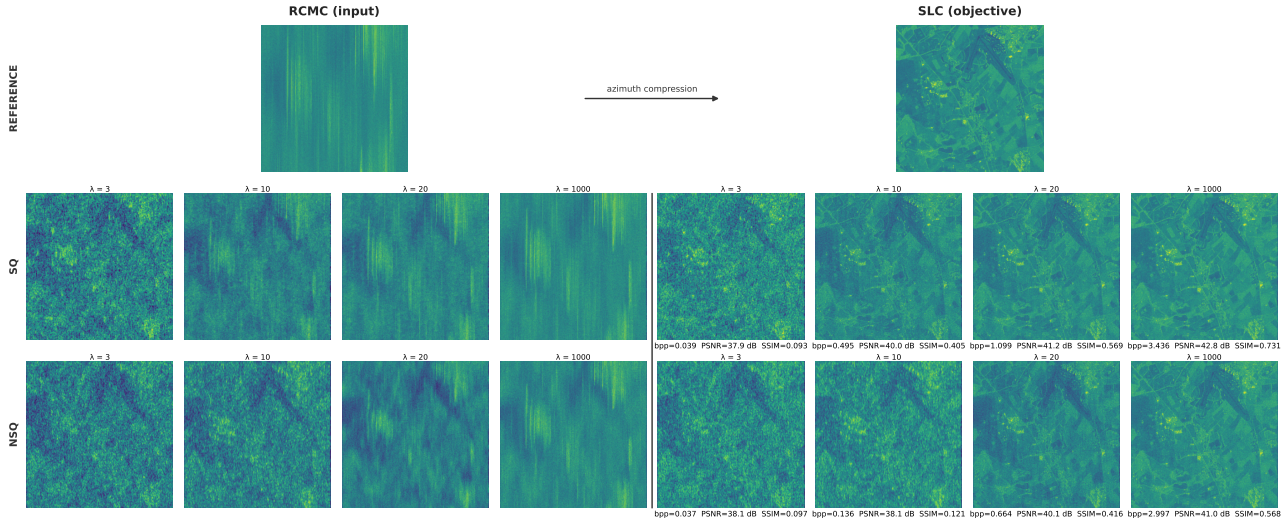

Figure 4 Visualization of the reconstructions: On the left $\widehat{\texttt{rcmc}}$, the direct output of the SH and SH-NSK models, on the right $\widehat{\texttt{az}}$, after azimuth compression. The top row shows the reference `rcmc` and `az` patches from Maya4.

artifacts from incomplete azimuth compression.

All models are trained and evaluated on a subset of the Maya4 dataset comprising $11{,}000$ patches, split into 64% training, 18% validation, and 18% testing sets. Each model is trained for up to 30 epochs with early stopping. Reconstruction quality is assessed using pixel-wise metrics such as Peak Signal-to-Noise Ratio (PSNR), perceptual metrics such as SSIM, and SAR-specific metrics including complex coherence and phase error. We evaluate the Scale Hyperprior (SH) architecture with two variants: standard square convolution kernels (3×3, 5×5), and Non-Square Kernels (NSK) with rectangular shapes in the azimuth direction (9×3, 15×5), denoted as SH-NSK.

Figure 3 presents Rate-Distortion (RD) performance of the proposed models alongside traditional codecs (JPEG, WebP, JPEG2000)[4]. To apply these codecs to complex-valued `rcmc` data, compression is performed independently on the real and imaginary components, effectively simulating grayscale images. Traditional codecs perform poorly in this setting. SSIM values remain below 0.1, indicating that the reconstructed SLC images lack meaningful structural information. Interestingly, PSNR increases at lower bitrates, which can be explained by the tendency of aggressive compression to produce smoother, more homogeneous reconstructions with reduced pixel-wise variance. This behavior highlights the limitations of PSNR in this context and underscores the need for perceptual and task-relevant metrics.

Comparing the LIC approaches, the standard SH architecture consistently outperforms the SH-NSK variant across most of the RD range. SH-NSK requires higher bitrates to reach comparable reconstruction quality. For instance, SH achieves visually coherent reconstructions at $\lambda = 10$ and below 0.5 bpp, whereas SH-NSK requires $\lambda = 20$ and approximately 0.7 bpp to reach similar quality. At higher bitrates, SH further improves upon SH-NSK by up to 2 dB in PSNR or 0.2 in SSIM. One possible explanation for SH-NSK's inferior performance is that, while asymmetric kernels are designed to better capture directional dependencies, they may introduce an imbalance in modeling range and azimuth features. In our setting, this could lead to suboptimal representation of fine-scale structures in the range dimension. From a SAR perspective, both models demonstrate partial preservation of phase information. At higher bitrates, complex coherence reaches values of approximately 0.95 and phase error remains around 0.04, indicating residual distortions that may limit the usability of the reconstructed data for downstream applications such as interferometry.

Visual inspection of the reconstructions (Figure 4) provides additional insight. As the rate increases, azimuth structures become sharper and more consistent with the expected SLC patterns. At lower bitrates, however, azimuth signatures appear shortened and distorted, resulting in RCMC reconstructions that resemble noisy, coarsely focused SLC images. We attribute this behavior to the limited receptive field of the networks, which restricts their ability to model long-range dependencies in the azimuth direction, although further investigation is required to confirm this hypothesis.

5 Conclusion

In this work, we proposed an end-to-end neural compression framework for intermediate SAR representations, with a focus on RCMC data. By integrating azimuth compression into the evaluation pipeline, we ensured that the reconstructed signals remain consistent with downstream SLC formation. In addition, we introduced a modified distortion loss that explicitly enforces coherence preservation.

Our results show that neural compression models can capture and reconstruct meaningful RCMC structures, en-

[4]Classical SAR-specific quantization schemes, such as BAQ or FDBAQ, are not included in this comparison, as they are typically designed for raw echo data and are not directly applicable to intermediate representations, such as RCMC, without modification. A quantitative comparison with such approaches is left for future work.

abling the recovery of interpretable SLC images after azimuth compression. The standard Scale Hyperprior architecture consistently outperforms its non-square kernel variant, achieving up to 0.76 SSIM, which highlights the importance of architectural design choices in this context.

Despite these promising results, limitations remain. While amplitude structures are reasonably well preserved, residual phase errors and incomplete coherence conservation may hinder the applicability of the reconstructed data to interferometric tasks.

Future work will focus on improving reconstruction fidelity, particularly in terms of phase preservation, and on benchmarking the proposed approach against traditional SAR compression schemes, such as BAQ-based methods. These directions will be essential to assess the practical relevance of neural compression for onboard SAR data processing.

6 Literature

[1] Reza Mohammadi Asiyabi, Andrei Anghel, Paola Rizzoli, Michele Martone, and Mihai 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*, pages 1787–1790, July 2023. doi: 10.1109/IGARSS52108.2023.10282287.

[2] Reza Mohammadi Asiyabi, Andrei Anghel, Adrian Focsa, Mihai Datcu, Michele Martone, Paola Rizzoli, and Ernesto Imbembo. On the use of JPEG2000 for SAR raw data compression. In *EUSAR 2024; 15th European Conference on Synthetic Aperture Radar*, pages 249–253, April 2024.

[3] Evert Attema, Ciro Cafforio, Michael Gottwald, Pietro Guccione, Andrea Monti Guarnieri, Fabio Rocca, and Paul Snoeij. Flexible Dynamic Block Adaptive Quantization for Sentinel-1 SAR Missions. *IEEE Geoscience and Remote Sensing Letters*, 7(4): 766–770, October 2010. ISSN 1558-0571. doi: 10.1109/LGRS.2010.2047242.

[4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end Optimized Image Compression, March 2017.

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, February 2018.

[6] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. CompressAI: A PyTorch library and evaluation platform for end-to-end compression research, November 2020.

[7] D. L. Bickel and A. W. Doerry. Using coherence as a quality measure for complex radar image compression. In *Radar Sensor Technology XXI*, volume 10188, pages 500–511. SPIE, May 2017. doi: 10.1117/12.2258351.

[8] Dirk Geudtner, Michel Tossaint, Malcolm Davidson, and Ramón Torres. Copernicus Sentinel-1 Next Generation Mission. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 874–876, July 2021. doi: 10.1109/IGARSS47720.2021.9554226.

[9] Nicola Gollin, Michele Martone, Ernesto Imbembo, Max Ghiglione, Stefan Knoll, Gerhard Krieger, and Paola Rizzoli. AI-BAQ: Deep Learning for Adaptive SAR Raw Data Quantization. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–20, 2025. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2025.3607089.

[10] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006. doi: 10.1126/science.1127647.

[11] David C. Lancashire, Bartholomew A. F. Barnes, and Stephen J. Udall. Block adaptive quantization, July 2001.

[12] Srikanth Mandapati, Ulrich Balss, and Helko Breit. Real Time Floating Point SAR Focusing on FPGA. In *EUSAR 2024; 15th European Conference on Synthetic Aperture Radar*, pages 60–65, April 2024.

[13] David Minnen, Johannes Ballé, and George D Toderici. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Majid Rabbani. JPEG2000: Image Compression Fundamentals, Standards and Practice. *Journal of Electronic Imaging*, 11(2):286, April 2002. ISSN 1017-9909, 1560-229X. doi: 10.1117/1.1469618.

[15] R.K. Raney, H. Runge, R. Bamler, I.G. Cumming, and F.H. Wong. Precision SAR processing using chirp scaling. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):786–799, July 1994. ISSN 1558-0644. doi: 10.1109/36.298008.