ISPRS Journal of Photogrammetry and Remote Sensing 220 (2025) 524–534

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

# Joint compression and despeckling by SAR representation learning

Joel Amao-Oliva [a],[*], Nils Foix-Colonier [b], Francescopaolo Sica [c]

[a] German Aerospace Center, Microwaves and Radar Institute, Münchener Str. 20, Weßling, 82234, Germany
[b] Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, 1 rue de la Noë, Nantes, 44000, France
[c] Department of Aerospace Engineering, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577, Neubiberg, Germany

ARTICLE INFO

*Keywords:*
Synthetic Aperture Radar (SAR)
Despeckling
Image compression
Machine learning
Self-supervised learning
Representation learning

ABSTRACT

Synthetic Aperture Radar (SAR) imagery is a powerful and widely used tool in a variety of remote sensing applications. The increasing number of SAR sensors makes it challenging to process and store such a large amount of data. In addition, as the flexibility and processing power of on-board electronics increases, the challenge of effectively transmitting large images to the ground becomes more tangible and pressing. In this paper, we present a method that uses self-supervised despeckling to learn a SAR image representation that is then used to perform image compression. The intuition that despeckling will additionally improve the compression task is based on the fact that the image representation used for despeckling forms an image prior that preserves the main image features while suppressing the spatially correlated noise component. The same learned image representation, which can already be seen as the output of a data reduction task, is further compressed in a lossless manner. While the two tasks can be solved separately, we propose to simultaneously training our model for despeckling and compression in a self-supervised and multi-objective fashion. The proposed network architecture avoids the use of skip connections by ensuring that the encoder and decoder share only the features generated at the lowest network level, namely the bridge, which is then further transformed into a bitstream. This differs from the usual network architectures used for despeckling, such as the commonly used Deep Residual U-Net. In this way, our network design allows compression and reconstruction to be performed at two different times and locations. The proposed method is trained and tested on real data from the TerraSAR-X sensor (downloaded from https://earth.esa.int/eogateway/catalog/terrasar-x-esa-archive). The experiments show that joint optimization can achieve performance beyond the state-of-the-art for both despeckling and compression, represented here by the *MERLIN* and *JPEG2000* algorithms, respectively. Furthermore, our method has been successfully tested against the cascade of these despeckling and compression algorithms, showing a better spatial and radiometric resolution, while achieving a better compression rate, e.g. a Peak Signal to Noise Ratio (PSNR) always higher than the comparison methods for any achieved bits-per-pixel (BPP) and specifically a PSNR gain of more than 2 dB by a compression rate of 0.7 BPP.

## 1. Introduction

Over the past decades, space-borne Synthetic Aperture Radar (SAR) imagery has proven to be extremely useful for a wide range of applications. With the recent increase of SAR missions from several space agencies around the world and from private companies, data storage and downlink are one of the major challenges in the ground segment of a mission. In addition, with the recent advances in on-board processing systems, both hardware and software, on-board SAR processing is becoming a reality and a desired solution not only for near real-time applications. As a result, the need for effective compression algorithms in the space segment is becoming more urgent.

SAR image compression is not a new task in remote sensing, nor a consequence of the incremental use of deep learning (DL) based processing of SAR data. On the contrary, since decades researchers dedicated efforts in SAR image compression, as seen in Baxter (1999) Mercier (2003). The use of the wavelet transform for compression is very common as shown in Zeng and Cumming (2001). Similarly, sparse representations have been used successfully for compression (Zhan et al., 2013). DL based image compression, on the other hand, is relatively new. Recent work on end-to-end optical image compression has shown that DL-based approaches can outperform conventional compression methods, such as JPEG, JPEG2000, WebP, and BPG, as

[*] Corresponding author.
*E-mail address:* joel.amao@dlr.de (J. Amao-Oliva).

https://doi.org/10.1016/j.isprsjprs.2024.12.016
Received 31 January 2024; Received in revised form 19 December 2024; Accepted 20 December 2024
Available online 14 January 2025

well as other machine learning (ML) approaches (Ballé et al., 2017, 2018). However, the applicability of these approaches to SAR data is limited due to the presence of speckle. The problem of speckle removal, or despeckling, is an extensively studied topic and is usually treated as a separate task. Despeckling has been tackled by various classical methods (Argenti et al., 2013), such as filtering in a transformed space, namely variational approaches, and the most recent non-local methods, among others. In recent years, several DL-based methods have been proposed that address the despeckle task in supervised and self-supervised frameworks. The most recent approach that has shown state-of-the-art performance is the MERLIN method (Dalsasso et al., 2022b), which exploits the independence between the real and imaginary parts of a single SAR image to set up a self-supervised framework for learning the underlying SAR reflectivity. As the same authors show in a follow-up paper, this approach can be effectively used to provide a learned representation of the SAR signal capable of supporting concurrent tasks such as segmentation and regression (Dalsasso et al., 2023).

In this paper, we address the task of joint compression and despeckling of SAR images by exploiting learned SAR representations. We show that the two tasks can help each other to obtain a clean image while preserving spatial and radiometric resolution and a corresponding compressed signal dimension that is reduced down to a fraction of a bit per pixel. We attribute this good result to the fact that both tasks independently aim at obtaining a representation of the SAR image that preserves the main signal features. Optimizing the network in a self-supervised manner and with a multi-objective loss function allows us to obtain the best performance in terms of despeckling and compression compared to the state-of-the-art. The concept that denoising can also promote additional tasks has already proven effective for natural images for the tasks of compression (Cheng et al., 2022) and segmentation (Buchholz et al., 2020), and for SAR data for segmentation and regression (Dalsasso et al., 2023). By exploiting the training strategy proposed in Dalsasso et al. (2022b), our methodology automatically takes into account the SAR signal statistics as well as its main characteristics for the purpose of improving SAR image compression. To show the importance of this last aspect, we further compare with the cascade of state-of-the-art despeckling and compression algorithms and show that the proposed approach can achieve better performance. Additionally, our contribution was to design an ad-hoc network architecture where the encoder and decoder components do not share any parameters. This is done to separate the compression process of the SAR image from its subsequent reconstruction. This departure from conventional network structures, such as the widely used Deep Residual U-Net for despeckling, guarantees the independence of the compression and reconstruction operations. As a result, this approach can be used for ground segment processing, such as data compression and subsequent reconstruction at a later time, as well as for the space segment, by separating the actual network architecture in two, the encoder part on board the platform and the decoder head on the ground. The resulting “sAr Despeckling And coMpression” (ADAM) framework has been trained and tested on real TerraSAR-X images and shows state-of-the-art performance for both despeckling and compression tasks.

## 2. Related work

In recent years, ML has shown remarkable results in various image processing tasks, including detection, classification, semantic segmentation, and denoising. Several application areas have also been explored in the SAR domain, such as land cover classification, detection, parameter inversion, despeckling, interferometric SAR (InSAR) data processing, and fusion of SAR and optical images (Fracastoro et al., 2021a). Due to the statistical characteristics of SAR images, classical DL models cannot be applied directly, but always need to be adapted to the peculiarities of the SAR signal. As a result, the most promising approaches are those that usually take into account the system parameters, such as geometry, acquisition modality and operating frequency, as well as the statistical properties of the signal to tailor the method to the specific SAR application.

Some of the most common pitfalls associated with the use of DL with SAR data can be summarized as follows (Zhu et al., 2021; Fracastoro et al., 2021b):

- The lack of ground truth is often solved by resorting to generating synthetic data in the framework of supervised learning. Implicitly, this is equivalent to simulating the image prior and the likelihood, with the consequence that any deviation from the real data will generate errors in the final result.
- The acquisition geometry greatly affects the SAR data, e.g. by producing effects such as layover and shadows that are related to a specific geometry. Therefore, data augmentation such as patch rotation, which is often required to ensure better DL model training, would not make much sense for SAR.
- Speckle statistics must be carefully considered. The signal-dependent and multiplicative nature of speckle noise complicates the use of classical DL methods. In addition, the signal-dependent nature of the speckle noise and the large dynamic range of the SAR lead to high noise variance. Often, homomorphic transformations are used to transform the multiplicative noise into an additive noise, which also implies to consider the new signal statistics in the homomorphic domain.

### 2.1. Despeckling

DL offers a variety of approaches for despeckling, leveraging different network architectures and training strategies. These methods have been reviewed in Denis et al. (2021) and explored in greater depth in Fracastoro et al. (2021b), which also discusses potential future directions. Furthermore, Dalsasso et al. (2020) investigates the impact of pre-training strategies and compares their benefits against end-to-end training approaches.

In this section, we introduce some of the main concepts of DL-based SAR despeckling, which often involves multiple steps of data preprocessing, diverse training strategies based on different types of data, and pre-trained networks. A key distinction in these methods lies between supervised and self-supervised methodologies.

Supervised training strategies have demonstrated strong performance in SAR despeckling tasks. Training a neural network in a supervised manner involves selecting a network architecture and training it on paired (input, reference) data. The training process minimizes the error between the network’s output and the reference image. A reference dataset is often achieved by data simulation, which offers extreme flexibility in training the network but, on the other hand, could introduce errors due to possible mismatches between simulated and real data. Alternatively, real data can be used through temporal multi-looking to generate a “ground truth” image (Mazza et al., 2021). This ground truth serves as reference for training, while the input is typically the detected backscatter of a single-look complex (SLC) SAR image. This approach is often referred to as “semi-supervised”, where user supervision is present but the training primarily relies on noisy real data. An example of this approach can be found in the SAR2SAR method (Dalsasso et al., 2021).

Self-supervised training strategies are particularly attractive because they do not require reference data, such as ground truth images or synthetic datasets. The primary advantage of these methods is their ability to avoid domain shifts between the training data and the operational data, a limitation often encountered in supervised approaches (Dalsasso et al., 2022a).

A recent self-supervised training approach introduced in Dalsasso et al. (2022b) exploits the statistical properties of SAR images. It builds on the observation that the real and imaginary parts of an SLC

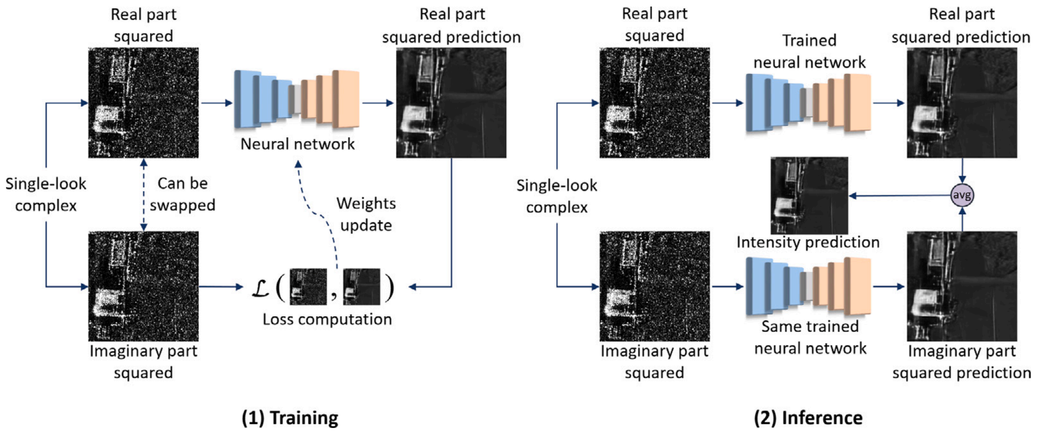


**Fig. 1.** (1) MERLIN training strategy; (2) Inference of the underlying reflectivity with a given network architecture.

SAR image are independent and identically distributed realizations of the same random process. This insight allows for a unique training scheme: the squared real part is used as the input to the network, while the squared imaginary part serves as reference (or vice versa). By leveraging this property, the network learns to despeckle SAR images without requiring external ground truth or synthetic training data.

One can decompose the received complex SLC as $\eta = Ae^{i\varphi} = a + ib,\ (a,b) \in \mathbb{R}^2$, where its pdf can be expressed as:

$$\begin{aligned} p_\eta(a+ib) &= \frac{1}{\pi s} e^{-|a^2+b^2|/s} \\ &= \frac{1}{\sqrt{2\pi}\sqrt{s/2}} e^{-a^2/s} \frac{1}{\sqrt{2\pi}\sqrt{s/2}} e^{-b^2/s}, \end{aligned} \tag{1}$$

with its real and imaginary parts identically distributed: $a \sim \mathcal{N}(0, s/2)$ and $b \sim \mathcal{N}(0, s/2)$. Here, $s$ represents the underlying reflectivity. The intensity image can be expressed as $I = Re^2 + Im^2$, where $Re^2 = a^2$ and $Im^2 = b^2$. The MERLIN approach, visually presented in Fig. 1, uses this property of SAR data to perform self-supervised learning, using as input $Re^2$ (or $Im^2$) and as a reference $Im^2$ (or $Re^2$) to estimate the SAR reflectivity. Given the function $f_\theta$, the trained model parametrized in $\theta$, at test time MERLIN applies twice the inference: once on $Re^2$ and once on $Im^2$ to obtain a final intensity estimate:

$$I = \frac{1}{2}\left(f_\theta(Re^2) + f_\theta(Im^2)\right). \tag{2}$$

### 2.2. Compression

Common DL-based approaches for image compression use autoencoder architectures. An autoencoder is a feedforward neural network with the same input and output shapes, along with two mirrored sets of layers, the encoder and decoder, connected by a generally smaller representation of the data corresponding to the latent space. When the goal is to accurately reproduce the input, the autoencoder finds a representation in a space that is usually compressed with respect to the original domain. The size of this latent space can be chosen, for example, to reduce data dimensionality. Considering this, one can compare the encoder and decoder to the transform and the inverse transform typically used in compression schemes such as transform coding. An example of a data flow in transform coding is shown in Fig. 2.

We will refer to the encoding-decoding operation as the analysis and synthesis transform, respectively, and denote them by $g_a(\cdot)$ and $g_s(\cdot)$. The input data, e.g., 2D images, is denoted by $x$, while its latent representation is denoted by $y = g_a(x)$. The reconstructed data estimated by the autoencoder is denoted by $\hat{x} = g_s(y) = g_s(g_a(x))$.

A variant of the autoencoder architecture, the variational autoencoder (VAE), was introduced in Kingma and Welling (2013), based on the idea of adding probabilistic constraints onto the latent space, considered as samples of distributions that are learned during training as well. Such distributions can be learned by minimizing a Kullback–Leibler divergence term (Goodfellow et al., 2016), which measures how different two distributions are, which is then added into the loss. Variational autoencoders have shown success in the literature (Ballé et al., 2017, 2018), presenting even better compression ratios than the state-of-the-art compression approaches. Compressing SAR images via variational autoencoders has also seen some success in recent years (Xu et al., 2022; Di et al., 2022). However, such architectures work directly on detected images. In addition, joint decompression and denoising approaches for optical images also had been explored in the literature with good performance (Cheng et al., 2022; Alves de Oliveira et al., 2022a), further supporting the idea of jointly despeckling and compressing SAR images. An end-to-end compression architecture utilizing VAE developed in Ballé et al. (2017) performs self-supervised compression, in which instead of computing the Kullback–Leibler divergence term to learn the distributions, the bitrate, i.e. the bits per pixel required to encode the information, is estimated from an entropy-based calculation, where the entropy model is assumed to adopt a parametric form in which its parameters are fitted to the data. This rate term, denoted by $R$, is then added to the loss function that needs to be minimized, where the smallest bitrate is achieved when the entropy is minimized if the model distribution is identical to the marginal (Ballé et al., 2018). To compress the data, the input of the arithmetic encoder must be quantized. However, the quantization function is not differentiable everywhere, which is unsuitable for backpropagation during the training phase. Therefore, instead of optimizing the original rate term, which should be the discrete entropy of the vector of quantization indices, the differential entropy is optimized (Ballé et al., 2016). This entropy is based on a continuous relaxation of the signal's density, which is obtained by convolving said density with an uniform distribution modeling quantization noise. This allows to train the neural network with a differentiable estimation of the rate, while the inference

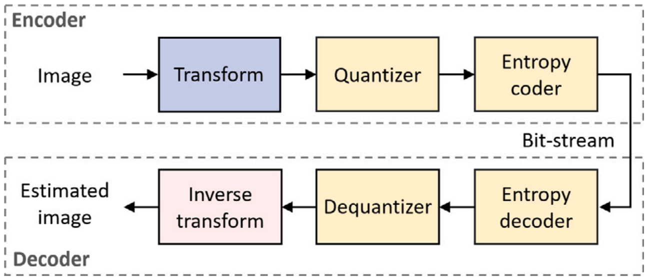


**Fig. 2.** Data flow usually found in transform coding.

will be performed with an actual quantization function. The other term in the loss function is the distortion $D$, which can be the mean squared error (MSE) or another distortion metric (Ballé et al., 2017). Since one wants to minimize both terms, the proposed loss function $J$ was formulated as a relaxed rate–distortion optimization problem by introducing a Lagrangian multiplier $\lambda$. With the same notation as in the previous section, and with $\hat{y}$ as a quantized version of $y$, the loss function can be defined as:

$$J(x, \hat{y}, \hat{x}) = R(\hat{y}) + \lambda D(x, \hat{x}). \tag{3}$$

Variations of $\lambda$ in Eq. (3) above will change the rate–distortion trade-off of the network trained with this loss. Indeed, if one gives more importance to the distortion, the network will try to minimize the distortion rather than the rate, resulting in more bits required to encode the compressed image.

The network introduced in Ballé et al. (2018) was initially made of convolutional layers with a particular activation function called generalized divisive normalization (GDN), with its "reciprocal" function being the inverse GDN (IGDN). The GDN function was defined and proved well-defined and invertible in Ballé et al. (2015) and adds nonlinearity into the trained models, achieving higher compression ratios while minimizing the compression distortion (Ballé et al., 2021; Ballé, 2018). It is important to note that, in the VAE proposed in Ballé et al. (2018), the training and predicting take two different operations:

- When the network performs a prediction, the input $x$ goes through the convolutional layers of the analysis transform, and $y = g_a(x)$ is obtained. Then the same latent representation is quantized into $\hat{y}$, thanks to a quantizer denoted by "Q". $\hat{y}$ is then encoded with binary arithmetic coding (Ballé et al., 2017; Langdon, 1984), working thanks to an entropy model created with the learned prior distributions of the latent space, in which the bitstream is obtained after lossless arithmetic encoding. After reception, this bitstream is decoded by an arithmetic decoder, which accesses the same entropy model as the arithmetic encoder. Then, the synthesis transform is applied to $\hat{y}$, yielding the DL model's estimate $\hat{x} = g_s(\hat{y})$.
- During training, $y$, the output of the analysis transform, is perturbed by a uniform noise into $\hat{y}$, to simulate the effect of quantization (Ballé et al., 2017). Then, a differentiable upper estimation of its bitrate is made by an entropy model and kept in memory to add it into the loss function, see rate $R$ in Eq. (3). The synthesis transform is then applied to $\hat{y}$, yielding $\hat{x} = g_s(\hat{y})$, which is used to compute the loss. The model weights are updated, including the prior, so that the model learns the latent space's distributions.

This model had been extended in Ballé et al. (2018) to include a hyperprior that is estimated by a side network, further increasing its performance compared to the original (Ballé et al., 2017) architecture. Therefore, it can be assimilated to a hierarchical Bayesian model (Kruschke and Vanpaemel, 2015; van de Schoot et al., 2021) that would be more robust to changes in statistics in the data.

## 3. Proposed ADAM framework

Current state-of-the-art algorithms for SAR image compression often overlook the fact that prior despeckling can lead to higher compression rates and to more efficient data transfer. Low-entropy signals, such as the deterministic reflectivity characterized by regular patterns and redundancy, are more compressible than high-entropy signals like speckle noise, which lack structure and exhibit random behavior.

Current ML-based methods consider despeckling and compression tasks separately. While this separation makes sense from a processing point of view, when the goal is to estimate and compress the resulting reflectivity, a joint approach that can despeckle and compress the reflectivity simultaneously can be seen as optimal. Although the combination of joint compression and denoising is not a novel idea, state-of-the-art algorithms such as in Alves de Oliveira et al. (2022b) are tailored either for optical satellite images and the denoising was obtained through supervised learning by simulating pairs of noisy and noise-free images or for natural images (Cheng et al., 2020).

### 3.1. Network architecture

The proposed framework uses the concept of self-supervision for SAR despeckling, with the idea of learning an image representation through deep neural network architectures capable of joint despeckling and compression. Therefore, the resulting approach and the proposed model differ from the literature, in that they rely on a previously unpaired architecture and training strategy. Moreover, the training and prediction steps are performed in two different ways to optimize the joint task, as shown in Figs. 3 and 5. The following subsections provide more details on the preprocessing required for the framework and the testing and training strategies. It should be noted that unlike the residual U-Net architecture presented in Dalsasso et al. (2022b), the variational autoencoder architecture chosen in this work does not allow skip connections, which are instrumental in restoring fine details. Given the constraints of the compression task, we opted for a variational autoencoder architecture augmented with residual blocks to avoid vanishing gradients during training. This architecture includes quantization, entropy coding/decoding of the latent space, and a side hyperprior variational autoencoder. This integrated design improves compression rates despite the need for an additional side bitstream for the prior scales, and provides a high degree of adaptability to varying input data.

### 3.2. Pre-processing of the data

Several preprocessing steps must be taken to avoid problems when using SAR data with neural networks. One concern is the peculiarity of SAR data, which has an intrinsically high dynamic range and possible correlation between real and imaginary parts (e.g., caused by processing the raw data with a squint). Therefore, logarithmic transformation, spectrum centering, and symmetrization have been introduced into the preprocessing flow. In addition, the preservation of point/strong scatterers also required special processing. Concerning ML-linked constraints, two additional operations were introduced: normalization and image partitioning into patches. These preprocessing steps are presented in the same order as they are used in the preprocessing chain.

- Symmetrization: It is necessary to ensure that the real and imaginary parts of SLC data are independent. As one will be used as the input of a neural network, while the other will be the reference on which the training will be done through a gradient descent. To enforce such independence, the preprocessing steps considered are spectrum centering and symmetrization, as described in Dalsasso et al. (2022b).
- Preservation of point-like scatterers: It is well known that strong scatterers do not follow the fully developed speckle model: these bright points exhibit deterministic behavior. Under these conditions, the hypothesis used to train the MERLIN model is no longer valid. To avoid possible radiometric inaccuracies in our estimation, we propose a modification of MERLIN's self-supervised training by modifying the input data. Specifically, we first identify those pixels in the amplitude image that have a value greater than 9 dB and assign the same value, equal to half the power of the signal, to both the real and imaginary parts. We also note that additional work on MERLIN (Meraoumia et al., 2023) has shown that, although the fully developed speckle hypothesis does not hold for strong scatterers, the algorithm does not introduce a bias in the estimation. However, we provide experimental evidence

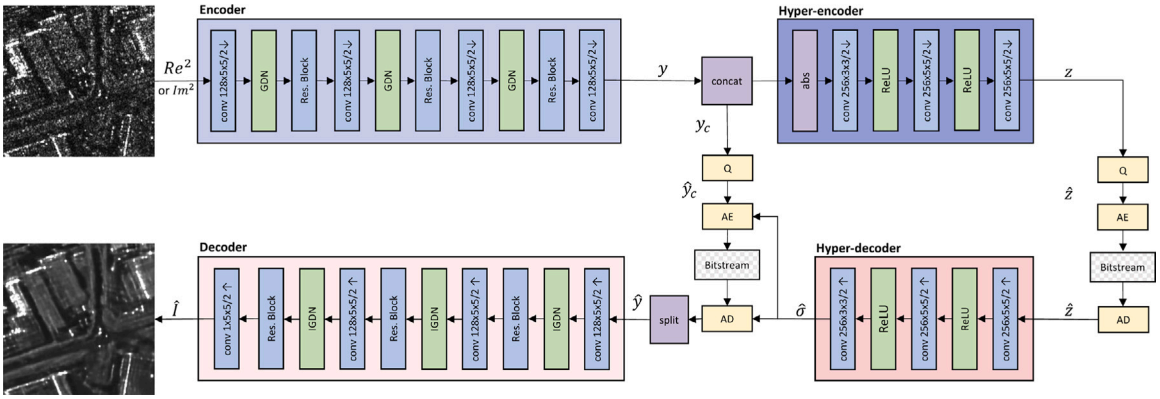


**Fig. 3.** Proposed network architecture with hyperprior. Q stands for quantization, AE and AD for the arithmetic encoders and decoders, respectively. The concatenation doubles the features to allocate the correct number of features needed at test time. The terms $y$, $y_c$, $\hat{y}_c$ refer to the estimated, joint concatenated, and quantized latent spaces, respectively, while $z$ and $\hat{z}$ are the estimated and quantized distributions resulting from the hyper-encoder.

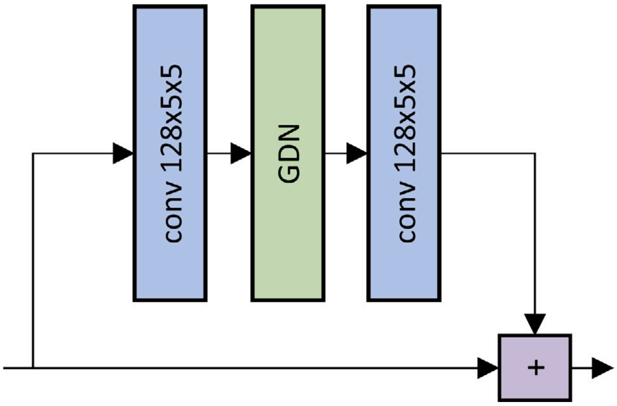


**Fig. 4.** Residual block used during training and testing as seen in Figs. 3 and 5. The goal of the residual block is to overcome the problem of vanishing gradients and make training deep networks easier and more effective (He et al., 2016), mainly thanks to the skip connections, which allow a more efficient backpropagation.

that the MERLIN approach has worse radiometric resolution for strong scatterers compared to SAR-BM3D and our approach. We show this in the experimental section by testing the preservation of peak intensity values of corner reflectors.

- Log transformation and normalization: To reduce the dynamic range of the input data, we convert real and imaginary parts to the log-domain and apply a fixed affine transformation to the $Re^2$ and $Im^2$ images.
- Split into patches: A common practice consists of splitting the input image into smaller square images, called "patches", which the neural network will process one by one, or batch by batch. In the experiments presented here, the patches are 256 × 256 pixels. Once all the patches of an image are processed, it is possible to reconstruct an image with the exact dimensions by aggregating them.

### 3.3. Training strategy

The network is trained with pairs of real and imaginary parts using the Noise2Noise approach proposed in MERLIN. The encoder–decoder architecture includes variational autoencoders and residual blocks. The training process includes careful hyperparameter selection to ensure optimal performance and robustness. A scale hyperprior is integrated into the architecture to improve compression efficiency. It estimates the parameter $\sigma$ of the probability distribution corresponding to each feature of the latent space. During training, these features are doubled by a concatenation operation of the latent space with itself. This allows to allocate the correct number of features for the real and imaginary parts at test time.

Eventually, quantization and entropy encoding/decoding are applied.

#### 3.3.1. Encoder/decoder

The encoder module, pictured in light blue in Fig. 3, consists of several blocks of convolutional layers, with a downsampling operation followed by GDN layers acting as activation layers, with a residual block (see Fig. 4) at the end, used to increase the networks receptive field and its compression-rate performance (Cheng et al., 2020). The output $y$ is the latent representation of input $Re^2$ or $Im^2$, which is concatenated with itself in the "concat" block, resulting in $y_c$. The concatenation operation is necessary to train the network to handle latent spaces twice the size of the input $Re^2$ or $Im^2$, since real and imaginary latent spaces must simultaneously undergo the same processing: encoding, concatenation, quantization, and transmission to the ground at test time, as shown in Fig. 5. Similarly, the Decoder module, shown in light pink in Fig. 3, reverts the process of the Encoder taking the estimated vector $\hat{y}$ as input and returns the estimated reflectivity of the $Re^2$ or $Im^2$ input. Note that the latent representations corresponding to the real and imaginary inputs are concatenated only for compression purposes. In the reconstruction stage, the two latent spaces are separated again and processed independently.

#### 3.3.2. Hyper-encoder/hyper-decoder

To further decrease the bitrate, a scale hyperprior was added to the network (Ballé et al., 2018) represented in the hyper-encoder/decoder blocks, in dark blue and dark pink, which are composed of convolutions and linear rectifiers. The hyper-encoder takes the concatenated latent space $y_c$ and then estimates its distributions, here indicated as $z$, which is then quantized, compressed, and transmitted as side information. On the receiver side, the estimated quantized $\hat{z}$ is fed into the hyper-decoder to estimate $\hat{\sigma}$, which is then used to decode $\hat{y}_c$.

#### 3.3.3. Loss function

The loss function optimizes the rate $R$ given a distortion term $D$. Thus, minimizing the loss means reducing the rate $R$ as much as possible, given an acceptable distortion $D$ of the image. The balance between rate and distortion can be set in the loss by a weight $\lambda$ as shown in Eq. (3). The rate $R$ is measured as the entropy of the latent space and $D$ is a measure of the quality of the reconstruction of the despeckled image. Therefore, the previous equation becomes:

$$\begin{aligned}\mathcal{L}(\hat{I}, Im) &= R + \lambda D(\hat{I}_k, Im_k)\\ &= R(\hat{y}) + R(\hat{z})\\ &+ \lambda \sum_k \frac{1}{2} \log \hat{I}_k + \exp\left(2 \log|Im_k| - \log \hat{I}_k\right),\end{aligned} \tag{4}$$

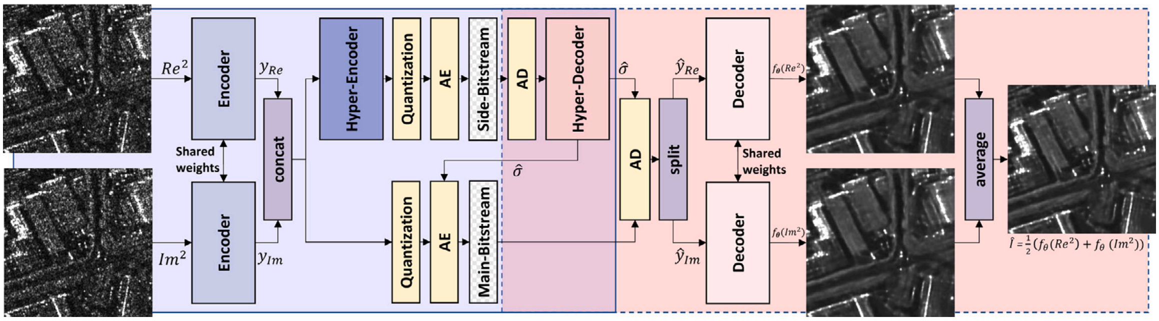


**Fig. 5.** Proposed ADAM framework: The left side, highlighted in blue in the diagram, represents the modules part of the transmitter, while the blocks on the right side, with a pink tone, represent the modules on the receiver side. The hyper-decoder appears in both the transmitter and receiver since the estimated scales $\hat{\sigma}$ must be known by the encoder and the decoder.

where $D$ is the despeckling loss and $R$ is the entropy of the latent spaces $\hat{y}$ and $\hat{z}$. The input $Re^2$ (or $Im^2$) and the estimated reflectivity $\hat{I}$ are log-transformed and normalized. While $\lambda$ represents the Lagrangian multiplier that handles the trade-off between compression rate and despeckling, with lower values of $\lambda$ resulting in higher compression rates.

The loss function considered corresponds to a training strategy in which the input of the network $Re^2$ yields the output reflectivity $\hat{I}$, while the reference used to perform a gradient descent is the squared imaginary part $Im^2$. To train a network on more examples, $Re^2$ and $Im^2$ may be randomly swapped, as shown as the input of the encoder in Fig. 3, resulting in the loss function $\mathcal{L}(\hat{I}, Re)$. The rate $R$ is the expected code length (bitrate) of the distributions of $y$ and $z$, which are minimized during training using the entropy models $p_{\hat{y}}(\hat{y})$ and $p_{\hat{z}}(\hat{z})$ (Ballé et al., 2018). The joint despeckling/distortion loss then corresponds to the sum over all pixels of the opposite of the log-likelihood of the marginal distributions, in log-scale (Dalsasso et al., 2022b).

### 3.4. Testing strategy

While the self-supervised MERLIN approach (Dalsasso et al., 2022b), utilizing the real/imaginary diversity of SAR data, lends itself well to a despeckling task, an obvious drawback when trying to apply it for compression is the fact that the bitrate is doubled when encoding both the real and imaginary parts since both are needed to estimate the final reflectivity. The proposed solution handles this issue by effectively using a single hyper-encoder and hyper-decoder, shown in Fig. 5, that is applied to a joint latent space $\hat{y}_c = \mathrm{concat}(y_{Re}, y_{Im})$, after splitting the complex SLC into real and imaginary parts. This latent space is then used to estimate the scales $\hat{\sigma}$ using during arithmetic encoding/decoding in the hyperprior. The estimation of the reflectivity is then given by averaging the estimations of the real and imaginary components and can be formalized as follows:

$$\hat{I} = \frac{1}{2}\left(f_\theta(Re^2) + f_\theta(Im^2)\right), \tag{5}$$

where $f_\theta(Re^2) = (\mathrm{dec}(\mathrm{split}(\hat{y}_c, Re))$ represents the decoded real component of the reflectivity, and $f_\theta(Im^2) = \mathrm{dec}(\mathrm{split}(\hat{y}_c, Im))$ represents the decoded reflectivity component from the imaginary part. The term $\mathrm{dec}(\cdot)$ represents the decoding operation, and $\mathrm{split}(\cdot)$ the split operation, which, when applied to the concatenated latent space $\hat{y}_c$, returns the estimated real $\hat{y}_{Re} = \mathrm{split}(\hat{y}_c, Re)$ and imaginary $\hat{y}_{Im} = \mathrm{split}(\hat{y}_c, Im)$ latent space.

## 4. Experimental results

In order to evaluate the performance of the proposed method with respect to the state-of-the-art, we performed three types of experiments aimed at testing different properties of the processed images. In particular, we tested the spatial resolution, radiometric resolution, and the preservation of strong scatterers. To this end, we use different performance metrics based on local image features that do not rely on the use of ground truth. Additionally, we build a synthetic ground truth to complement the results. Summarizing, we will show the results on:

- Despeckling metrics that do not require a ground truth
- Joint despeckling and compression metrics on a synthetic ground truth
- Preservation of the radiometric resolution of strong scatterers

To properly assess the performance with the state-of-the-art, we compare the proposed method against the following methods:

- SAR-BM3D (Parrilli et al., 2012): state-of-the-art despeckling algorithm among non-DL-based methods.
- MERLIN (Dalsasso et al., 2022b): DL-based state-of-the-art despeckling method.
- MERLIN + JPEG2000 (Taubman et al., 2002): a cascade application of these two algorithms functions as the baseline for joint despeckling and compression. It uses MERLIN to estimate the reflectivity followed by a JPEG2000 compression.

### 4.1. Despeckling metrics

To detect biases in the estimated reflectivities, several indicators can be utilized over homogeneous areas:

- Mean of image (MoI): Should be preserved during filtering.
- Mean of ratio (MoR): The mean of the ratio image should be as close to 1 to indicate good radiometric preservation.
- Variance of ratio (VoR): If the MoR is then close to 1, a VoR less than 1 usually indicates under-smoothing, and a value greater than 1 indicates over-smoothing (Di Martino et al., 2013).
- Ratio image: The ratio image should contain only speckle after filtering. The presence of structures in the ratio image can be a sign of overfiltering.

Speckle reduction over homogeneous areas is usually measured in terms of the equivalent number of looks (ENL):

$$\mathrm{ENL} = \frac{\mu_r^2}{\sigma_r^2}, \tag{6}$$

where $\mu_r$ represents the mean value and $\sigma_r$ is the standard deviation.

Table 1 summarizes the results on despeckling metrics over the homogeneous area in Fig. 7(a) highlighted in red. Additionally, in Fig. 6 we show the ratio images over the area surrounding the German Aerospace Center (DLR) in Neustrelitz, Germany. The figures show the result of the processed images in the first row and the ratio images in the second row.

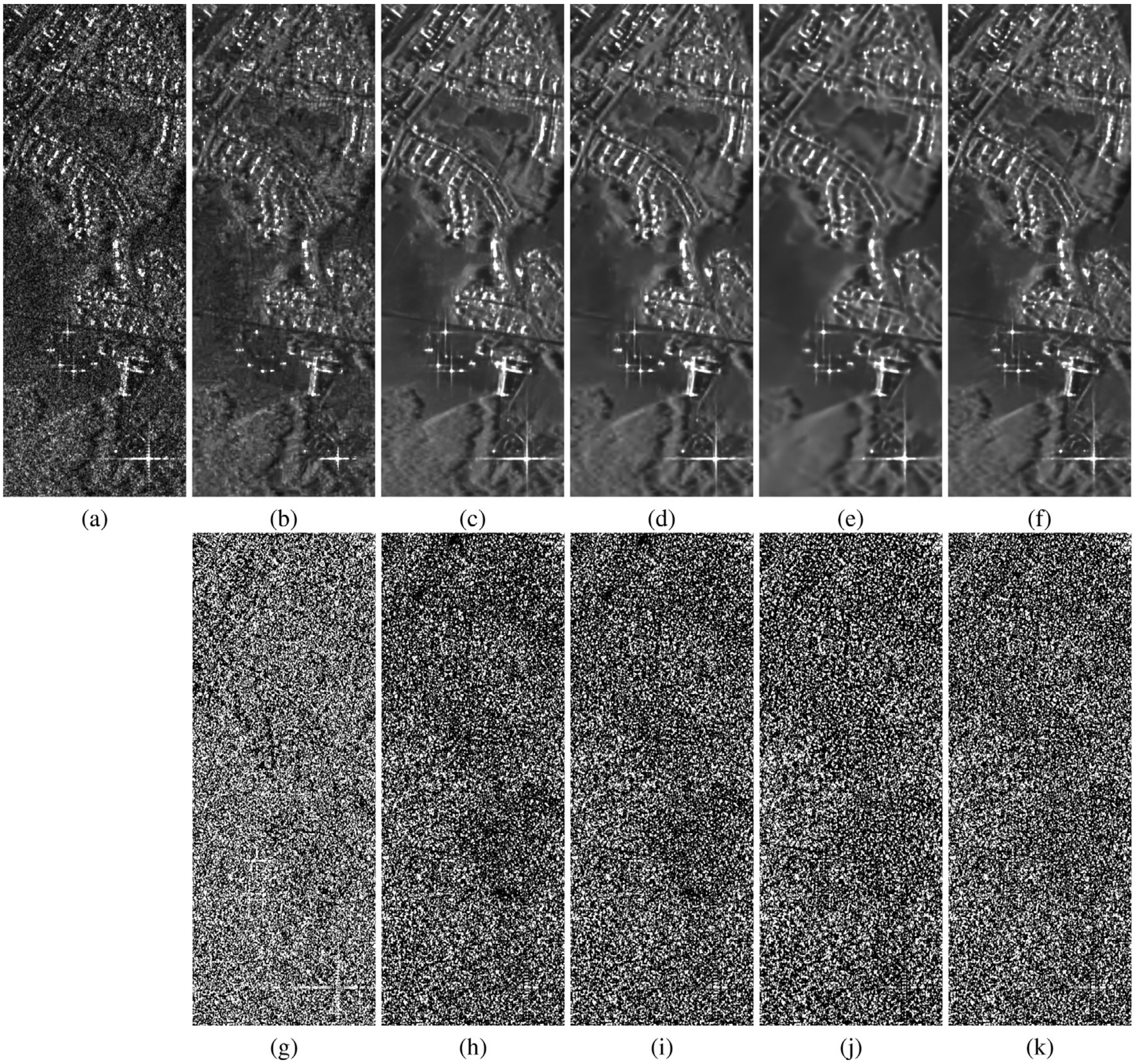


**Fig. 6.** (a)–(f) Zoomed images over the Neustrelitz area corresponding to (a) noisy image, (b) SAR-BM3D, (c) MERLIN, (d) MERLIN+JPEG2000, (e) ADAM with $\lambda = 10$ and (f) ADAM with $\lambda = 90$. The images below show the corresponding ratio images.

**Table 1**
Despeckling metrics over homogeneous area.

| | MoI | MoR | VoR | ENL |
|---|---|---|---|---|
| Noisy image | 0.178 | – | – | 1 |
| ADAM $\lambda = 10$ | 0.166 | **0.98** | **0.91** | **332.17** |
| ADAM $\lambda = 40$ | **0.176** | 0.96 | 0.85 | 204.90 |
| ADAM $\lambda = 90$ | 0.170 | 0.98 | **0.86** | 214.16 |
| MERLIN | 0.172 | 0.96 | 0.81 | 118.18 |
| MERLIN+JPEG2000 | 0.173 | 0.97 | 0.85 | **406.03** |
| SAR-BM3D | **0.177** | **0.99** | 0.63 | 10.88 |

### 4.2. Joint despeckling and compression metrics

To evaluate the performance of our approach with respect to the cascade of state-of-the-art despeckling and compression algorithms, we need to build a synthetic ground truth and test a quantitative performance metric at different compression rates. To build the synthetic ground truth, we followed the multitemporal approach proposed in Vitale et al. (2021). Specifically, we averaged over time 21 TerraSAR-X co-registered intensity images in the HH polarization. It is worth noting that while it would have been possible to apply the same method to a larger set of images to obtain a better reference image, the larger time span would have increased the probability of incorporating temporal changes and thus producing a less reliable ground truth image. Using this ground truth, we measure the distortion of joint despeckling and compression methods via the Peak Signal to Noise Ratio (PSNR), which measures the total distortion introduced by the compression approach when compared to an uncompressed and ideally noise-free reference image. For simplicity, we will refer to the resulting synthetic ground truth shown in Fig. 7(b) as the *Hamburg* test site. The PSNR is then defined as follows:

$$\mathrm{PSNR}(A, \hat{A}) = 10 \log_{10} \frac{\max(A)}{\sqrt{\mathrm{MSE}(A, \hat{A})}}, \tag{7}$$

where MSE represents the mean-square-error estimator, $\hat{A}$ is the estimated amplitude, and $A$ is the reference *Hamburg* amplitude.

For this experiment, we also consider a *noisy baseline* computed by applying JPEG2000 compression directly to the noisy image. We included it to also consider the inherent ability of a compression algorithm to perform despeckling. The results of the rate distortion performance based on the PSNR and the resulting bits-per-pixel (BPP) over the *Hamburg* test image are shown in Fig. 8.

### 4.3. Preservation of the radiometric resolution of strong scatterers

To evaluate the effects of despeckling and compression over strong scatterers, we considered an area around the DLR site in Neustrelitz,

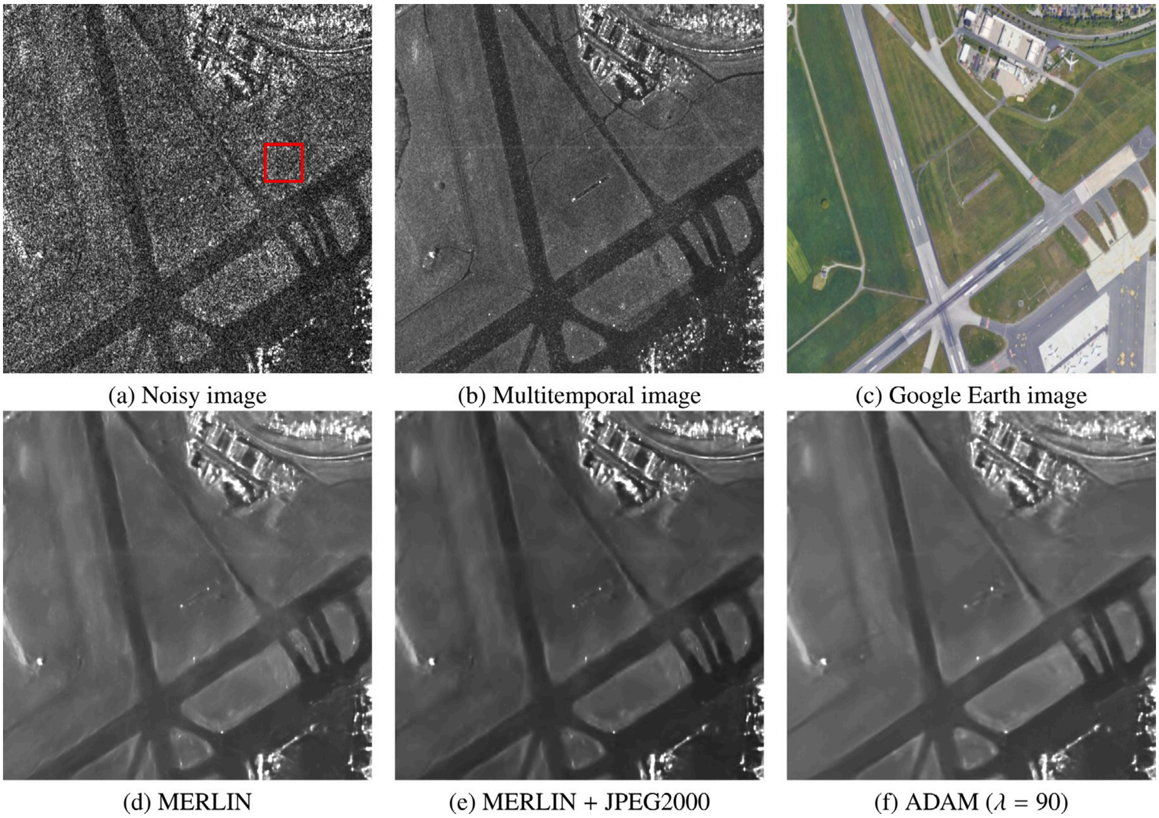


**Fig. 7.** Results over the Hamburg scene: (a) Noisy image with a red square highlighting the homogeneous area used to estimate several performance metrics. (b) The multitemporal image obtained via averaging. (c) Google Earth image used for visual inspection. (d) MERLIN estimated reflectivity and (e) MERLIN+JPEG2000 estimated reflectivity. (f) Our approach.

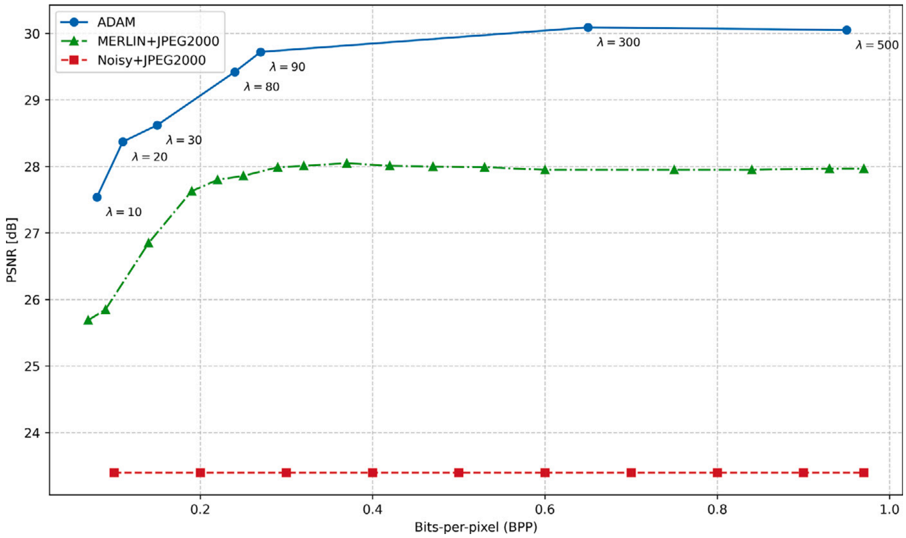


**Fig. 8.** Rate–distortion performance measured by the PSNR of the network tested on the Hamburg scene, with two additional baselines (Noisy+JPEG2000 and MERLIN+JPEG2000) for comparison.

Germany, where several corner reflectors are placed for calibration purposes, as visible in Fig. 9. The computed average of slant range profiles over these corner reflectors are shown in Fig. 10.

To assess the radiometric preservation in these areas, we used two intensity contrast measures (Di Martino et al., 2013):

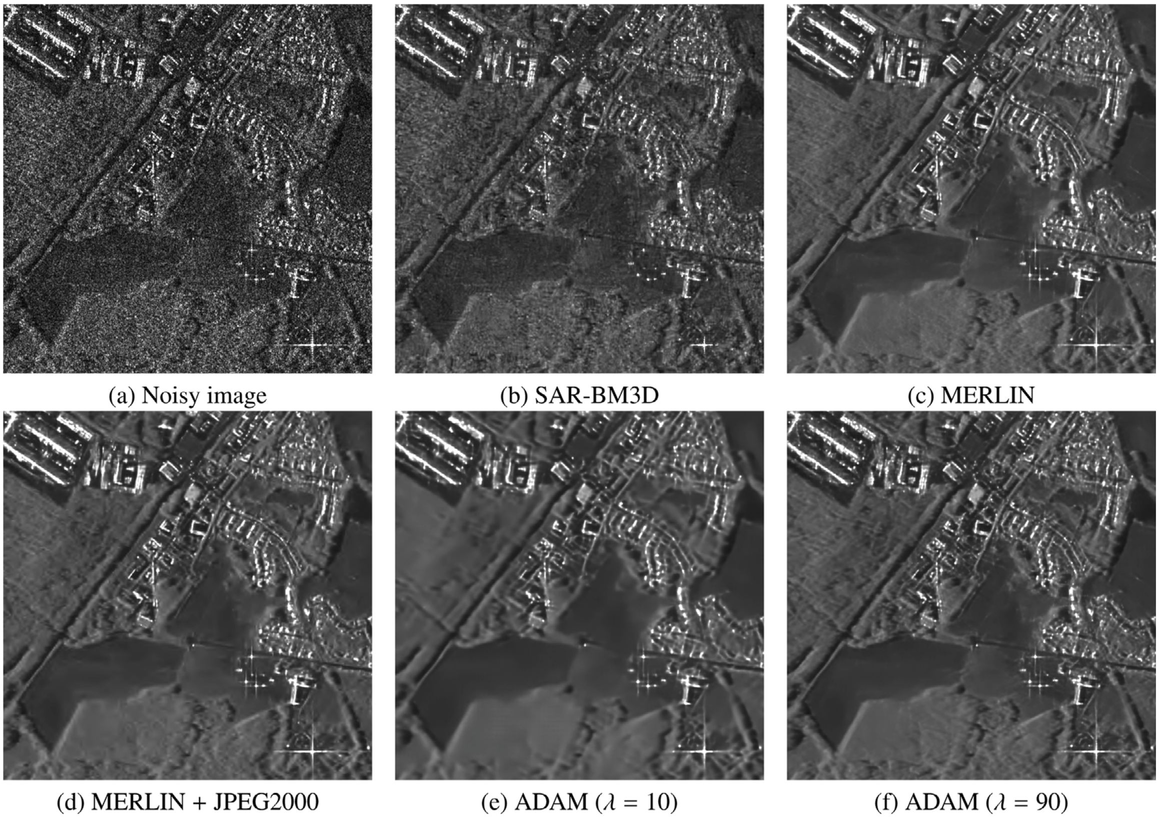
$$C_{\mathrm{nn}} = 10 \log_{10} \frac{r_{\mathrm{CF}}}{r_{\mathrm{NN}}}, \tag{8}$$

$$C_{\mathrm{bg}} = 10 \log_{10} \frac{r_{\mathrm{CF}}}{r_{\mathrm{BG}}}, \tag{9}$$

where:

- $r_{\mathrm{CF}}$ is the reflectivity observed in the corner reflector.
- $r_{\mathrm{NN}}$ is the average reflectivity observed in the eight connected nearest pixels of the corner reflector.
- $r_{\mathrm{BG}}$ is the average reflectivity of the background.

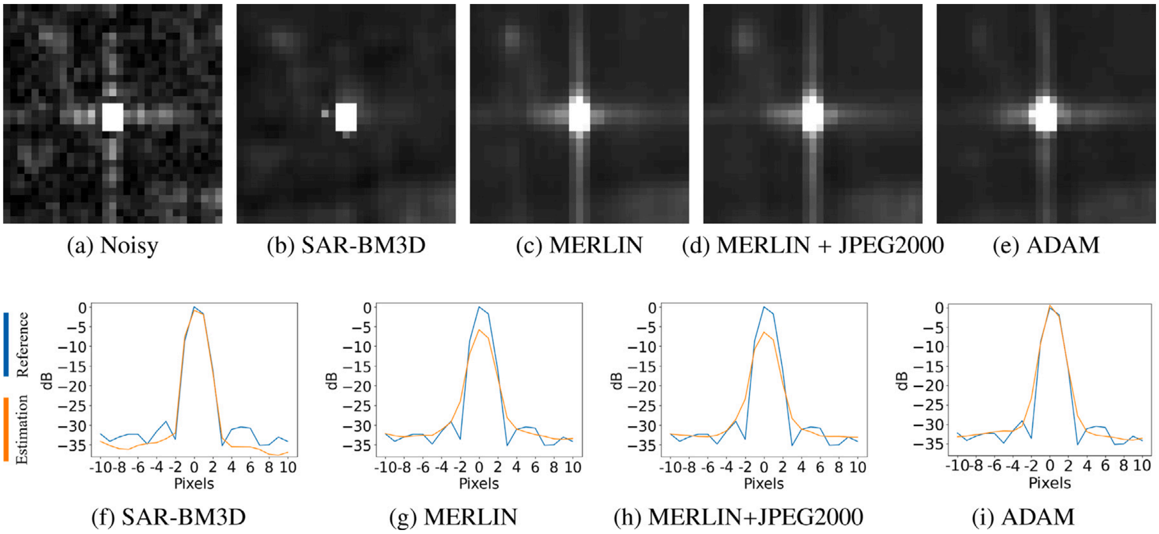
Eq. (8) measures the contrast between the corner reflector and its surrounding pixels, and Eq. (9) measures how well the bright points of the image are preserved compared to the background reflectivity



(a) Noisy image (b) SAR-BM3D (c) MERLIN

(d) MERLIN + JPEG2000 (e) ADAM ($\lambda$ = 10) (f) ADAM ($\lambda$ = 90)

**Fig. 9.** Results over the Neustrelitz scene, containing several corner reflectors used to evaluate the radiometric performance of the proposed method.



(a) Noisy (b) SAR-BM3D (c) MERLIN (d) MERLIN + JPEG2000 (e) ADAM

(f) SAR-BM3D (g) MERLIN (h) MERLIN+JPEG2000 (i) ADAM

**Fig. 10.** Zooms over the estimated reflectivities of one of the corner reflectors of the Neustrelitz area. (f)–(i) Range profiles of the competing methods and our approach ($\lambda$ = 10). Figures (b)–(c), and their corresponding profiles, do not include any compression, while (d) and our method (e) present compressed results.

level. Additionally, we further compute for each method their absolute deviations with respect to the noisy reference:

$$\Delta C_{\mathrm{nn}} = |C_{\mathrm{nn}} - C_{\mathrm{nn}}^{\mathrm{ref}}| \tag{10}$$

$$\Delta C_{\mathrm{bg}} = |C_{\mathrm{bg}} - C_{\mathrm{bg}}^{\mathrm{ref}}| \tag{11}$$

The results of these metrics are summarized in Table 2.

## 5. Discussion

The results obtained demonstrate the validity of the proposed approach, showing performance equal to or superior to the state-of-the-art in terms of both spatial and radiometric resolution. In addition, we tested a special case of radiometric resolution preservation: strong scatterers, e.g. corner reflectors, whose reconstruction is especially critical. Comparisons were made in terms of despeckling capability, but also in terms of compression, and we have shown that even compressing the data to a fraction of a single bit per pixel would still result in despeckling performance close to or superior to the state-of-the-art.

**Table 2**
Corner reflector results.

| | $C_{\mathrm{nn}}$ [dB] | $\Delta C_{\mathrm{nn}}$ [dB] | $C_{\mathrm{bg}}$ [dB] | $\Delta C_{\mathrm{bg}}$ [dB] |
|---|---|---|---|---|
| Noisy image | 26.75 | | 42.54 | |
| ADAM $\lambda = 10$ | 28.25 | 1.50 | **42.51** | **0.03** |
| ADAM $\lambda = 40$ | 29.11 | 2.36 | 43.15 | 0.61 |
| ADAM $\lambda = 90$ | 28.21 | 1.45 | **42.26** | **0.29** |
| MERLIN | **25.69** | **1.06** | 35.90 | 6.64 |
| MERLIN+JPEG2000 | 25.43 | 1.32 | 35.66 | 6.88 |
| SAR-BM3D | **25.63** | **1.12** | 41.60 | 0.94 |

In Fig. 6 we visually assess the speckle suppression capabilities of the compared methods. The MERLIN and MERLIN+JPEG2000 results show residual noise patterns that correlate with image structures, suggesting an overestimation of reflectivity over urban areas. The results of our method with both $\lambda = 10$ and $\lambda = 90$ show no obvious presence of errors in the estimation of reflectivity. In fact, in the ratio images in Figs. 6(j) and 6(k), there is no trace of any structure or obvious pattern, further confirming the ability of the proposed method to suppress speckle while preserving all details.

The results of the despeckling metrics over the Hamburg area are shown in Table 1, which summarizes the MoI, MoR, VoR, and ENL values. For MoI, the closer the results were to the original noisy image mean, the better. For MoR and VoR, the results closest to 1 are considered the best, indicating proper filtering. For ENL, the higher the value, the better the performance. For this experiment, the two best results are highlighted in black. The results on MoI, MoR, and VoR metrics show that the proposed method performs best, by preseving well radiometric resolution and avoiding introducing biases. Conversely, the highest ENL was achieved using the MERLIN+JPEG2000 method. If we pair this result with the visual inspection of Fig. 6(d), we can ascribe the higher ENL values to oversmoothing, which is an unwanted effect. This is likely due to the inherent tendency of cascade approaches to oversmooth images.

The preservation of radiometric resolution has been further tested by visual inspection on the Neustrelitz image in Fig. 9. In this image we analyzed the behavior on different corner reflectors. Additionally, we plot the range profile of the corner reflector in the middle of the image as shown in Fig. 10. Our method shows the best radiometric resolution preservation, with SAR-BM3D being the second best algorithm. The results of the MERLIN+JPEG2000 baseline profile in Fig. 10(h) show a decrease in the image's dynamic range due to the compression performed by JPEG2000. The peak value remains the same as the MERLIN result in Fig. 10(g). We further computed the metrics $C_{\mathrm{nn}}$ and $C_{\mathrm{bg}}$ to complement the performance evaluation with a quantitative metric on corner reflectors. The results, shown in Table 2, also include the absolute deviations of these metrics for each method with respect to the noisy reference (Eqs. (10) and (11)). The two best results are those that show minimal distortion with respect to the noisy image, i.e. the lowest $\Delta$, and are highlighted in bold. We analyzed three different corner reflectors within the Neustrelitz scene. Fig. 10 zooms in on a single corner reflector, with the resulting range profiles underneath. The results associated with the MERLIN+JPEG2000 baseline were obtained with approximately 0.3 BPP. In the case of the $C_{\mathrm{nn}}$ metric, our approach with $\lambda = 90$ has a $\Delta C_{\mathrm{nn}}$ value of 1.45 dB. MERLIN, on the other hand, a $\Delta C_{\mathrm{nn}}$ value on the same corner reflector of 1.06 dB. The ADAM ($\lambda = 10$) result has shown a slightly higher total distortion in $\Delta C_{\mathrm{nn}}$ compared to the MERLIN+JPEG2000 baseline. However, this ADAM result was achieved with a bitrate of approximately 0.1 BPP, which is close to 66% less than the bitrate used in the MERLIN+JPEG2000 baseline for this case. For the $C_{\mathrm{bg}}$ metric, all results related to the proposed method resulted in higher dB values, indicating a higher suppression of the background reflectivity compared to the original noisy image. However, it is worth noting that higher $C_{\mathrm{bg}}$ values do not necessarily mean better preservation of point-like scatterers, but rather a contrast augmentation w.r.t. neighboring background pixels. In this regard, our approach with $\lambda = 10$ and $\lambda = 90$ performed the best.

We additionally tested rate-distorsion ratios for all the methods. To do that we trained our network several times with different $\lambda$ values. This analysis is useful to test the overall joint despeckling and compression performance. The rate-distorsion graphs are shown in Fig. 8. Based on these results, we found an optimal value for $\lambda$ of 90, which allows to obtain a BPP of around 0.3 and a PSNR value of about 1.5 dB and 6 dB higher with respect to the MERLIN+JPEG2000 and noisy baselines, respectively. The poor performance of the noisy baseline is expected given that speckle noise as well as the large dynamic range of SAR data lead to a higher entropy.

The proposed methodology has proven to be the best over the entire set of performance metrics presented. Nevertheless, we observed a variation in performance across the metrics for different $\lambda$ values, suggesting that the choice of $\lambda$ should be carefully considered depending on the desired performance. We also expect the need for fine-tuning or retraining and a new tuning of the hyperparameter $\lambda$ when applying the algorithm to other types of SAR data, which may differ in both radiometric and spatial characteristics. In the light of the results obtained, we can affirm that the superior performance of our method is due to the simultaneous optimization of the network for the compression and despeckling tasks, which together produce a better SAR image representation.

## 6. Conclusions

In this paper we presented a novel and effective approach to address the challenges associated with SAR data processing, particularly in terms of compression and despeckling. Leveraging the self-supervised machine learning paradigm, specifically the Noise2Noise framework used in the MERLIN despeckling approach, our proposed method employs an encoder–decoder architecture optimized to achieve both high speckle suppression and high compression rate. In particular, the network architecture employed, with separate encoder and decoder components for compression and reconstruction, differs from conventional designs. This design choice ensures the independence of these two operations, allowing their separate execution at different times and locations. The joint optimization strategy, performed in a self-supervised and multi-objective manner, results in superior performance. The resulting latent representation of SAR data is analyzed using a hyperprior encoder to improve compression performance. Our experiments, conducted on real TerraSAR-X data, show that the proposed method outperforms state-of-the-art techniques in both despeckling and compression tasks. Furthermore, the results demonstrate the flexibility of our model, which is able to preserve more detail at the expense of bitrate and vice versa by simply varying the hyperparameter $\lambda$. The proposed loss function allows for adaptability in the latent space, opening avenues for future work to extend the model's capabilities to additional tasks, such as segmentation, in a multitask learning fashion.

## CRediT authorship contribution statement

**Joel Amao-Oliva:** Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Nils Foix-Colonier:** Writing – original draft, Validation, Software, Methodology, Investigation. **Francescopaolo Sica:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors would like to thank the TerraSAR-X European Space Agency (ESA) archive for providing access to the TerraSAR-X data used in this work. We would also like to thank the anonymous reviewers for their accurate and meticulous comments that largerly helped to improve the quality of this manuscript.

## References

Alves de Oliveira, V., Chabert, M., Oberlin, T., Poulliat, C., Bruno, M., Latry, C., Carlavan, M., Henrot, S., Falzon, F., Camarero, R., 2022a. Satellite image compression and denoising with neural networks. IEEE Geosci. Remote. Sens. Lett. 19, 1–5.

Alves de Oliveira, V., Chabert, M., Oberlin, T., Poulliat, C., Bruno, M., Latry, C., Carlavan, M., Henrot, S., Falzon, F., Camarero, R., 2022b. Satellite image compression and denoising with neural networks. IEEE Geosci. Remote. Sens. Lett. 19, 1–5. http://dx.doi.org/10.1109/LGRS.2022.3145992.

Argenti, F., Lapini, A., Bianchi, T., Alparone, L., 2013. A tutorial on speckle reduction in synthetic aperture radar images. IEEE Geosci. Remote. Sens. Mag. 1 (3), 6–35. http://dx.doi.org/10.1109/MGRS.2013.2277512.

Ballé, J., 2018. Efficient nonlinear transforms for lossy image compression. In: 2018 Picture Coding Symposium. pp. 248–252.

Ballé, J., Chou, P.A., Minnen, D., Singh, S., Johnston, N., Agustsson, E., Hwang, S.J., Toderici, G., 2021. Nonlinear transform coding. IEEE J. Sel. Top. Sign. Proces. 15 (2), 339–353.

Ballé, J., Laparra, V., Simoncelli, E.P., 2015. Density modeling of images using a generalized normalization transformation. arXiv:1511.06281.

Ballé, J., Laparra, V., Simoncelli, E.P., 2016. End-to-end optimization of nonlinear transform codes for perceptual quality. In: 2016 Picture Coding Symposium. PCS, IEEE, pp. 1–5.

Ballé, J., Laparra, V., Simoncelli, E.P., 2017. End-to-end Optimized Image Compression. arXiv:1611.01704.

Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N., 2018. Variational image compression with a scale hyperprior. arXiv:1802.01436.

Baxter, R.A., 1999. SAR image compression with the Gabor transform. IEEE Trans. Geosci. Remote. Sens. 37 (1), 574–588.

Buchholz, T.-O., Prakash, M., Schmidt, D., Krull, A., Jug, F., 2020. DenoiSeg: joint denoising and segmentation. In: European Conference on Computer Vision. Springer, pp. 324–337.

Cheng, Z., Sun, H., Takeuchi, M., Katto, J., 2020. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7939–7948.

Cheng, K.L., Xie, Y., Chen, Q., 2022. Optimizing image compression via joint learning with denoising.

Dalsasso, E., Denis, L., Muzeau, M., Tupin, F., 2022a. Self-supervised training strategies for SAR image despeckling with deep neural networks. In: 14th European Conference on Synthetic Aperture Radar. EUSAR, URL https://hal.telecom-paris.fr/hal-03589245.

Dalsasso, E., Denis, L., Tupin, F., 2021. SAR2SAR: A semi-supervised despeckling algorithm for SAR images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 4321–4329. http://dx.doi.org/10.1109/JSTARS.2021.3071864.

Dalsasso, E., Denis, L., Tupin, F., 2022b. As if by magic: Self-supervised training of deep despeckling networks with MERLIN. IEEE Trans. Geosci. Remote Sens. 60, 1–13. http://dx.doi.org/10.1109/TGRS.2021.3128621.

Dalsasso, E., Rambour, C., Denis, L., Tupin, F., 2023. Learning a versatile representation of SAR data for regression and segmentation by leveraging self-supervised despeckling with MERLIN.

Dalsasso, E., Yang, X., c Denis, L., Tupin, F., Yang, W., 2020. SAR image despeckling by deep neural networks: from a pre-trained model to an end-to-end training strategy. Remote Sens. 12 (16), 2636. http://dx.doi.org/10.3390/rs12162636.

van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M.G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J., Yau, C., 2021. Bayesian statistics and modelling. Nat. Rev. Methods Primers 1 (1), 1–26.

Denis, L., Dalsasso, E., Tupin, F., 2021. A review of Deep-Learning Techniques for SAR image restoration. In: 2021 IEEE International Geoscience and Remote Sensing Symposium. IGARSS, pp. 411–414. http://dx.doi.org/10.1109/IGARSS47720.2021.9555039.

Di, Z., Chen, X., Wu, Q., Shi, J., Feng, Q., Fan, Y., 2022. Learned compression framework with pyramidal features and quality enhancement for SAR images. IEEE Geosci. Remote. Sens. Lett. 19, 1–5. http://dx.doi.org/10.1109/LGRS.2022.3155651.

Di Martino, G., Poderico, M., Poggi, G., Riccio, D., Verdoliva, L., 2013. Benchmarking framework for SAR despeckling. IEEE Trans. Geosci. Remote Sens. 52 (3), 1596–1615.

Fracastoro, G., Magli, E., Poggi, G., Scarpa, G., Valsesia, D., Verdoliva, L., 2021a. Deep learning methods for synthetic aperture radar image despeckling: An overview of trends and perspectives. IEEE Geosci. Remote. Sens. Mag. 9 (2), 29–51. http://dx.doi.org/10.1109/MGRS.2021.3070956.

Fracastoro, G., Magli, E., Poggi, G., Scarpa, G., Valsesia, D., Verdoliva, L., 2021b. Deep learning methods for synthetic aperture radar image despeckling: An overview of trends and perspectives. IEEE Geosci. Remote. Sens. Mag. 9 (2), 29–51. http://dx.doi.org/10.1109/MGRS.2021.3070956.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational Bayes. arXiv:1312.6114.

Kruschke, J.K., Vanpaemel, W., 2015. Bayesian estimation in hierarchical models. Oxf. Handb. Comput. Math. Psychol. 279–299.

Langdon, G.G., 1984. An introduction to arithmetic coding. IBM J. Res. Dev. 28 (2), 135–149.

Mazza, A., Scarpa, G., Verdoliva, L., Poggi, G., 2021. Impact of training set design in CNN-based SAR image despeckling. In: 2021 IEEE International Geoscience and Remote Sensing Symposium. IGARSS, pp. 415–418. http://dx.doi.org/10.1109/IGARSS47720.2021.9553022.

Meraoumia, I., Dalsasso, E., Denis, L., Abergel, R., Tupin, F., 2023. Multitemporal speckle reduction with self-supervised deep neural networks. IEEE Trans. Geosci. Remote Sens. 61, 1–14. http://dx.doi.org/10.1109/TGRS.2023.3237466.

Mercier, G., 2003. Reflectivity estimation for SAR image compression. IEEE Trans. Geosci. Remote. Sens. 41 (4), 901–906.

Parrilli, S., Poderico, M., Angelino, C.V., Verdoliva, L., 2012. A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage. IEEE Trans. Geosci. Remote Sens. 50 (2), 606–616. http://dx.doi.org/10.1109/TGRS.2011.2161586.

Taubman, D.S., Marcellin, M.W., Rabbani, M., 2002. JPEG2000: Image compression fundamentals, standards and practice. J. Electron. Imaging 11 (2), 286–287.

Vitale, S., Ferraioli, G., Pascazio, V., 2021. Analysis on the building of training dataset for deep learning SAR despeckling. IEEE Geosci. Remote. Sens. Lett. 19, 1–5.

Xu, Q., Xiang, Y., Di, Z., Fan, Y., Feng, Q., Wu, Q., Shi, J., 2022. Synthetic aperture radar image compression based on a variational autoencoder. IEEE Geosci. Remote. Sens. Lett. 19, 1–5. http://dx.doi.org/10.1109/LGRS.2021.3097154.

Zeng, Z., Cumming, I.G., 2001. SAR image data compression using a tree-structured wavelet transform. IEEE Trans. Geosci. Remote Sens. 39 (3), 546–552.

Zhan, X., Zhang, R., Yin, D., Huo, C., 2013. SAR image compression using multiscale dictionary learning and sparse representation. IEEE Geosci. Remote. Sens. Lett. 10 (5), 1090–1094.

Zhu, X.X., Montazeri, S., Ali, M., Hua, Y., Wang, Y., Mou, L., Shi, Y., Xu, F., Bamler, R., 2021. Deep learning meets SAR: Concepts, models, pitfalls, and perspectives. IEEE Geosci. Remote. Sens. Mag. 9 (4), 143–172. http://dx.doi.org/10.1109/MGRS.2020.3046356.