Super-Resolution of Passive Microwave Satellite Retrievals Using Deep Learning: Exploring Deterministic and Probabilistic Model Architectures

Alessio Maurizio Canclini

(Dated: May 23, 2025)

Passive microwave satellite retrievals play a key role in observing Earth's cryosphere, yet some datasets are limited by coarse spatial resolution. This study evaluates deep learning-based single-image super-resolution (SISR) techniques to enhance such data, comparing a convolutional neural network (CNN), autoencoder (AE), variational autoencoder (VAE), generative adversarial network (GAN), and transformer-inspired attention models. Among them, the autoencoder REDNet performs best, achieving a mean test PSNR of 39.3 for vertical polarization (34.8 for horizontal) and SSIM of 0.867 (0.854), substantially outperforming the baseline values of 33.7 (31.2) PSNR and 0.791 (0.769) SSIM. Attention-based models show potential but are hindered by high memory demands and extended training times. The CNN and AE architectures benefit more from increased model width than from additional depth, and the VAE extension of REDNet enables probabilistic output with only minor degradation in performance. All models show reduced performance during melt season, highlighting sensitivity to environmental conditions. These results demonstrate the potential of deep learning for enhancing passive microwave observations and suggest directions for further research.

I. INTRODUCTION

Over the past decades, satellite observations have rapidly become indispensable tools to study and understand the impacts of climate change. In this context, passive microwave satellite remote sensing now provides over 40 years of continuous Arctic sea ice observations with the release of datasets such as the sea ice concentration climate data records by Lavergne *et al.* [1]. Sea ice plays a crucial role in the Arctic system by regulating the physical interactions between the atmosphere and ocean [2, 3], and acts as a key indicator of Arctic amplification; the fact that the Arctic is warming faster than the rest of the world [4]. These changes in high latitude regions are extremely important as the transition of sea ice areas becoming open ocean, drastically changes surface albedo and consequently radiative absorption. Daily passive microwave satellite retrievals since 1979 form one of the most important climate datasets, enabling long-term monitoring of these trends [5].

Figure 1 shows a polar map projection with passive microwave brightness temperatures over the Arctic Ocean for low post summer and high winter sea ice extent. Here we see the top of Norway in the lower right corner, Russia in the upper right, Canada and Alaska in the upper left, and Greenland in the lower left.

Despite the invaluable temporal coverage provided by passive microwave climatologies, their coarse spatial resolution remains a significant limitation. Sea ice exhibits complex and dynamic behaviour, including viscous and plastic deformation, which gives rise to fine-scale features such as cracks, leads, polynyas, and ridges. These features are critical, as they influence local ocean-atmosphere interactions and modulate the energy balance in the Arctic. However, data from the Special Sensor Microwave Imager (SSM/I), with a spatial resolution of approximately 25 km, are insufficient to resolve such detailed structures.

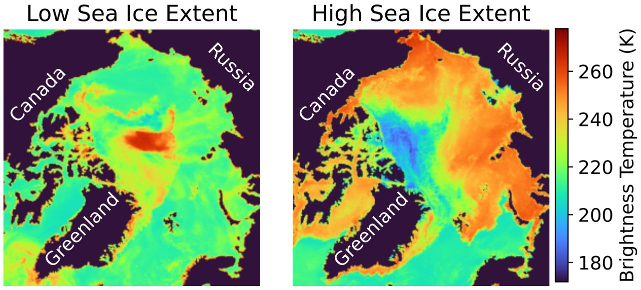

FIG. 1. Polar map projection of the Arctic region showing brightness temperatures from passive microwave satellite retrievals (AMSR) over ocean and sea ice. The maps depict conditions for low sea ice extent on September 15, 2020 (post-summer) and high sea ice extent on March 15, 2020 (mid-winter). Geographic references include Norway in the lower right, Russia in the upper right, Canada and Alaska in the upper left, and Greenland in the lower left corner.

To address this limitation in sea ice climatology, we investigate the application of machine learning (ML)-based Single Image Super-Resolution (SISR) techniques to enhance the spatial resolution of existing sea ice retrievals. Specifically, we leverage higher-resolution (12.5 km) sea ice products from the Advanced Microwave Scanning Radiometer (AMSR) sensor family, which provides $\sim$ 20 years of overlapping data with SSM/I since its launch in 2002.

This study evaluates the performance of various deep learning (DL) architectures in generating AMSR-quality passive microwave scenes from low-resolution SSMI inputs. Specifically, we explore a convolutional neural network (CNN), an autoencoder (AE), a variational autoencoder (VAE), a generative adversarial network (GAN), and two transformer-inspired attention-based models.

In the following, Section II provides the theoretical background and principles of SISR, along with descriptions and core principles of the deep learning architectures employed. This section concludes with the performance metrics used, details of the dataset, and model training procedures. Section III presents the results and discussion, covering both deterministic and probabilistic model performance as well as seasonal variability. Finally, Section IV summarizes the key findings and outlines directions for future work.

II. THEORY AND METHODS

A. Passive Microwave Remote Sensing

Any given material emits electromagnetic radiation as a function of its emissivity and temperature. Microwave radiometers are passive sensors which detect the amount of microwave radiation emitted from a surface. For aerial measurements, such sensors can be mounted on spacecrafts, air planes or satellites to measure the naturally emitted microwave radiation from the earths surface, also referred to as the radiance, and typically recorded as brightness temperature [6]. Brightness temperature measured in different bands of the microwave spectrum can be used to derive physical quantities such as wind speed, atmospheric water vapour, soil moisture, sea surface temperature and salinity, and sea ice concentration and type [5]. Notably, unlike thermal infrared radiation, microwave emission is not as strongly related to the materials temperature. Emissivity plays a greater role and is a function of the dielectric properties of the material. For a comprehensive overview of passive microwave remote sensing see Spreen and Kern [6]. We focus on the 37 GHz band as it is one of the frequencies which exhibits a range of brightness temperatures making it possible to discriminate between water and ice, as well as different ice types.

B. Single-Image Super-Resolution

SISR [7] is the task of generating a high-resolution (HR) image from a single low-resolution (LR) image depicting the same scene. The observed LR image is typically modelled as the result of a convolution between the unknown high-resolution image $\boldsymbol{y}$ and the blur kernel $\boldsymbol{K}$, followed by a downsampling operation with a scale factor s, and the addition of an independent noise term $\boldsymbol{n}$. This can be expressed as:

$$\boldsymbol{x} = (\boldsymbol{y} * \boldsymbol{K}) \downarrow_s + \boldsymbol{n} \tag{1}$$

where $*$ denotes convolution, $\downarrow_s$ represents the downsampling operator, and n accounts for noise. Solving Equation 1 is a notoriously challenging problem as any LR input may have many plausible HR solutions, making it an extremely ill-posed problem. Algorithms for SISR can be split into three main categories. Interpolation-based methods [8][9] are very quick and relatively simple, but also show the lowest accuracy results. Reconstruction-based SR methods [10][11] leverage prior knowledge to constrain the solution space, enabling sharp and flexible details. However, their performance deteriorates with larger scale factors and is often computationally expensive. Lastly, we have learning-based methods, which are the focus of this study. These methods use ML algorithms and achieve impressive performance and fast computation. To achieve this, such models require sufficient training examples to analyse statistical relations between the low- and high-resolution images.

C. Architectures and Core Principles

Subsequent sections will introduce SISR approaches using a variety of DL methods as well as the architecture's foundational principles. We consider both deterministic and probabilistic DL architectures. Deterministic models, such as CNNs, AEs, and transformers, learn fixed mappings from LR inputs to HR outputs without inherent uncertainty estimation. Probabilistic models like VAEs and most GANs incorporate stochasticity to model data distributions, allowing for diverse HR outputs that may help capture uncertainties. It should be noted that details have been omitted due to the variety of models, and the reader is referred to the original papers for a comprehensive description of each architecture. All model implementations are available at https://github.com/Alessimc/FYS5429.

1. *Convolutional Neural Network*

Discrete convolution is a fundamental operation in deep learning that applies small, learnable filters to extract spatial patterns from input data. Mathematically, a two dimensional convolution $\boldsymbol{Z} = \boldsymbol{I} * \boldsymbol{K}$ is defined as:

$$Z[i,j] = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} I[i-k, j-l] K[k,l]$$

Here $\boldsymbol{I}$ is the two dimensional input and $\boldsymbol{K}$ is a two dimensional filter, often called kernel. Brackets denote the indexing of matrix elements. The summation limits $-\infty$ to $+\infty$ conceptually assume an infinitely extended input, but in practice, $\boldsymbol{I}$ and $\boldsymbol{K}$ are nonzero only within a finite region. Therefore, this infinite summation is implemented as a summation over a finite number of elements.

Furthermore, the majority of ML libraries actually implement cross-correlation and more efficient algorithms exist for typical kernel sizes of 1 × 1, 3 × 3 and 5 × 5 [12]. Since convolution reduces the spatial dimensions of the output, padding is commonly applied to maintain the input size. Zero-padding, where the input is surrounded by zeros, is the most common approach and ensures that feature maps retain their resolution across layers.

Inspired by the human visual cortex, which is composed of several layers, extracting increasingly complex features, the first CNN was proposed by LeCun *et al.* [13] in 1989. CNNs have since shown incredible performance, especially in computer vision tasks such as image classification [14], object detection [15] and face recognition [16]. Relevant features are essential for any ML task and whilst many neural networks (NNs) require these as input, CNNs are able to learn features from raw data. These features are extracted through convolutional operations, where small filters (kernels) slide over the input like a moving window, capturing spatial patterns. This enables sparse interactions, as each filter processes only a local region, reducing computational cost. Parameter sharing further improves efficiency by applying the same filters across different locations, enhancing generalization. Additionally, CNNs exhibit equivariance to translation, meaning patterns remain recognizable regardless of their position, making them highly effective for vision tasks [17].

The chosen CNN architecture is based on the Super-Resolution Convolutional Neural Network (SRCNN) implementation proposed by Dong *et al.* [18]. This was the first published deep learning method for Super-Resolution (SR) and became very popular following its impressive results. It is a three-layer network using the ReLu activation function between layers, and the following filter sizes: $64 \times 1 \times 9 \times 9$, $32 \times 64 \times 1 \times 1$ and $1 \times 32 \times 5 \times 5$. These non-linear transformations result in patch extraction, non-linear mapping and reconstruction. Here, the model is optimised using Adam [19] and the Mean Squared Error (MSE) loss function. Specifically, given a collection of N training sample pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, where $\boldsymbol{x}_i$ is a LR image and $\boldsymbol{y}_i$ is the HR version as the ground truth, we minimise

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \|F(\boldsymbol{x}_i;\theta) - \boldsymbol{y}_i\|_2^2,$$

where θ are trained weights represented by the convolutional kernels. $F(\boldsymbol{x};\theta) = \hat{\boldsymbol{y}}$ is the deterministic mapping the model learns to generate HR outputs $\hat{\boldsymbol{y}}$.

CNNs are powerful architectures by themselves, but more importantly play a key role as fundamental building blocks for some of the more complex ML architectures to follow.

2. *Autoencoder*

The AE was first introduced in 1985 as a form of NN designed to compress and reconstruct data by encoding it into a lower-dimensional latent representation [20]. Since then, AEs have evolved significantly through architectural improvements such as the integration of CNNs for spatial feature extraction and the adoption of probabilistic frameworks like Variational Bayes Inference, resulting in VAEs. These developments have enabled AEs to serve in diverse applications, including image generation, anomaly detection, and data compression. A particularly influential application is denoising, where AEs are trained to reconstruct clean data from noisy inputs [21]. A typical denoising AE applies noise to the input and learns to reconstruct the clean version. SISR can be viewed as a specific form of this problem, where the LR image serves as a noisy or degraded input, and the HR image is treated as the target output.

An AE consists of two components: an encoder $\boldsymbol{h} = f(\boldsymbol{x})$, which maps the input data into a latent space, and a decoder $\boldsymbol{y} = g(\boldsymbol{h})$, which reconstructs the expected output. The functions f and g are typically composed of multiple non-linear layers, such as fully connected or convolutional layers, with the latter being especially effective for image data. Since the dimensionality of $\boldsymbol{h}$ is usually smaller than that of $\boldsymbol{x}$, the encoder is tasked with extracting the most important features. This latent space is analogous to the subspace found by Principal Component Analysis (PCA), which projects data onto directions of maximum variance [12]. However, unlike a linear method such as PCA, AEs can learn non-linear transformations through deep architectures and activation functions, making them more expressive, especially for high-dimensional and structured data like images.

In this work, we adopt the Residual Encoder-Decoder Network (REDNet) architecture proposed by Mao *et al.* [22], which features a symmetric structure of convolutional layers for encoding and deconvolutional layers for decoding. Pooling layers are omitted to preserve spatial details essential for SISR. Training deep AEs can be challenging due to vanishing gradients and the difficulty of reconstructing fine details through many layers. To mitigate this, REDNet introduces skip connections between corresponding layers in the encoder and decoder, inspired by the residual learning concept introduced in Residual Network (ResNet) [23]. These connections allow both gradients and information to bypass intermediate layers, thereby stabilizing training and facilitating the preservation of high-frequency image details.

Rather than learning a direct mapping $\hat{\boldsymbol{y}} = F(\boldsymbol{x};\theta)$, residual networks model the residual and add it to the input:

$$\mathcal{F}(\boldsymbol{x};\theta) = \boldsymbol{y} - \boldsymbol{x} \Rightarrow \hat{\boldsymbol{y}} = \mathcal{F}(\boldsymbol{x};\theta) + \boldsymbol{x}.$$

This formulation simplifies the learning problem by focusing on modeling only the difference between the input and output, which is often small in SISR tasks.

Mao *et al.* [22] propose three REDNet variants: REDNet10, REDNet20, and REDNet30. These are made up of 5, 10, and 15 convolutional–deconvolutional layer pairs, respectively. All variants use 3×3 kernels and 128 filters by default, with skip connections introduced every second layer. Unlike many modern AEs that aggressively downsample inputs using stride-2 convolutions or pooling layers, REDNet downsamples only once in the first layer ($416 \rightarrow 208$) and maintains the spatial resolution until the final upsampling layer. This design choice preserves per-pixel information, which is critical for SISR. As for the SRCNN, training was performed using the Adam optimiser [19] and MSE loss, now to learn weights for the encoder and decoder.

3. *Transformer*

Although the underpinning idea of convolutions has been extremely successful, it is important to note that there exist other alternatives. The transformer is a neural network architecture using the principle of self-attention which developed from attention-based Recurrent Neural Networks (RNNs) as models showed even better performance when recurrent layers were removed [12]. At its core, the idea is that attention should help identify which features in the data are especially worth focusing on. The original attention mechanism was first introduced for language translation using RNNs [24]. In the sequential RNN, cross-attention has been used to link two modules, the encoder and decoder. Conceptually this is somewhat similar to the previously introduced skip connections as it facilitates information flow across distant parts of the network, although its purpose is to dynamically focus on relevant input features at each decoding step, rather than merely preserving and reusing feature representations from earlier layers.

In contrast to the original attention mechanisms in RNNs which link the encoder and decoder, the transformer first introduced by Deviyani and Singh [25] is built on self-attention, capturing dependencies between input elements.

We will first consider basic self-attention without any tunable parameters. Consider an input sequence $\{\boldsymbol{x}_i\}_{i=0}^{T}$, then the self-attention output is computed as

$$\boldsymbol{z}_i = \sum_{j=0}^{T} \alpha_{ij} \boldsymbol{x}_j,$$

where $\boldsymbol{z}_i$ becomes a context-aware embedding vector and α_{ij} are the attention weights. These are computed as

$$\alpha_{ij} = \mathrm{softmax}(\boldsymbol{x}_i^T \boldsymbol{x}_j).$$

Here each element in the attention weight matrix is the dot product between the current input element and another element, then normalized using the softmax function.

The transformer uses more advanced scaled dot-product attention which includes learnable parameters enabling it to optimise the attention weights. This requires three additional weight matrices, $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$. These project the inputs sequence into query, key and value sequences:

$$\begin{aligned} \text{Query: } \boldsymbol{q}_i &= \boldsymbol{Q}\boldsymbol{x}_i, \\ \text{Key: } \boldsymbol{k}_i &= \boldsymbol{K}\boldsymbol{x}_i, \\ \text{Value: } \boldsymbol{v}_i &= \boldsymbol{V}\boldsymbol{x}_i. \end{aligned}$$

Now the attention weights are computed as

$$\alpha_{ij} = \mathrm{softmax}\left(\frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{m}}\right),$$

where m is typically the length of $\boldsymbol{k}$ to ensure that the Euclidean length of the weight vectors is approximately the same range. These attention weights are then used to compute the final output as a weighted sum of the value vectors. The model optimizes the parameters in the projection matrices $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ (and those in the feed-forward layers), allowing it to learn task-specific feature interactions.

In practice, transformers use multi-head self-attention, which is repeating the scaled dot-product attention computation multiple times in parallel and combining the results. Scaled dot-product attention can be considered as a single attention head. These heads capture dependencies in different parts of the input, similar to how multiple kernels produce multiple channels in a CNN. One of the great advantages of multiple heads is that there are no dependencies and these can all be computed in in parallel.

The application of transformers to image-like data was introduced by Dosovitskiy *et al.* [26] with the Vision Transformer (ViT), showing that we are not limited to the use of CNNs. However, ViTs compute self-attention globally across the entire image, which leads to quadratic computational complexity with respect to image size and limits their ability to capture local context, which CNNs do naturally. To address these limitations, the Shifted windows (Swin) transformer [27] was proposed. It improves efficiency by dividing the image into non-overlapping windows (e.g., 7×7 patches) and computing self-attention within each window. This strategy reduces the complexity to linear with image size and preserves local contextual information. Furthermore, Swin introduces shifted windows to enable cross-window interactions, combining the benefits of both local processing and hierarchical representation.

Liang *et al.* [28] subsequently proposed SwinIR, specifically focusing on image restoration. This model is applied to SISR of passive microwave satellite retrievals to explore a pure transformer alternative to other typically CNN based architectures.

One of the great advantages of transformers; that many computations can occur in parallel, can also result as a training constraint due to the memory required with the high pixel resolution of image data. This has motivated hybrid architectures which leverage the proven strength of CNNs with the hopefully added value of attention mechanisms. One such model is the Efficient Long-Range Attention Network (ELAN) for image super-resolution [29]. Rather than applying full self-attention globally or even within fixed windows, ELAN uses efficient local attention blocks that selectively model long-range dependencies while maintaining low computational overhead. This design allows it to retain the inductive biases and efficiency of CNNs, while also benefiting from the dynamic feature weighting of attention. As a result, ELAN offers a compelling balance between performance and resource efficiency, making it particularly well-suited for high-resolution super-resolution tasks where pure transformer models may be prohibitively expensive to train.

To accommodate memory constraints when working with high-resolution passive microwave data, both SwinIR and ELAN were implemented in reduced configurations. For SwinIR, the model was scaled down by lowering the embedding dimension to 24, reducing the number of attention heads in each stage to four, and using a smaller attention window size of four. For ELAN, the number of attention blocks was reduced to 24, the internal channel dimensions were decreased to 60, and smaller attention window sizes of two, four, and eight were used instead of the larger defaults. For ELAN this matches the ELAN-light implementation proposed by Zhang *et al.* [29]. These configurations made it feasible to train the models on 416 × 416 resolution samples within available GPU memory. Both models were trained using the Adam optimizer and MSE loss.

4. Variational Autoencoder

One of the drawbacks of regular AEs as well as other methods discussed so far, is that they are not probabilistic, that is, they do not define a probability distribution. Previous methods define a deterministic mapping such that there is only one HR output for each LR input, not capturing the possible variability and uncertainty tied to the super resolution task. A VAE is a generative model introduce by Kingma *et al.* [30] and can be viewed as a probabilistic version of the traditional AE. Instead of just encoding an input to a fixed point in latent space, it encodes it to a probability distribution over the latent space and then samples from this distribution to construct the output. Since new HR images $\hat{\boldsymbol{y}}$ are now sampled from a distribution, the VAE can generate many different HR outputs for any given LR input.

To introduce this probabilistic ability into the AE, we cannot just randomly sample from the latent space, as it is very poorly structured and there is no guarantee that points near an images latent representation will resemble the original. To achieve a nicely organized latent space the encoder and decoder components are modified from the original AE.

Our goal in a VAE is to generate new data from a distribution $p_\theta(\boldsymbol{x})$ representing our dataset, although we do not fully know this distribution and only have access to the samples in the training set. Additionally we introduce the latent distribution $p_\theta(\boldsymbol{h})$ representing latent variables. Here θ are the parameters governing the distribution. These distributions exist in separate spaces and therefore require mappings between them. The posterior distribution $p_\theta(\boldsymbol{h}|\boldsymbol{x})$ gives the probability that a latent variable $\boldsymbol{h}$ is generated by a given image $\boldsymbol{x}$. The other mapping is the likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{h})$, which given a latent $\boldsymbol{h}$ tells us the probability of reconstructing an image $\boldsymbol{x}$. By sampling latent variables from the posterior distribution that were likely generated by images in the original data distribution, one can effectively generate new samples.

However, there are a few further complications. We do not know $p_\theta(\boldsymbol{h})$ and therefore assume a prior distribution such as a Gaussian. Furthermore, $p_\theta(\boldsymbol{h}|\boldsymbol{x})$ is required. Since we do not know the true posterior, it is approximated:

$$p_\theta(\boldsymbol{h}|\boldsymbol{x}) \approx q_\phi(\boldsymbol{h}|\boldsymbol{x}).$$

A typical choice is again a Gaussian $\mathcal{N}(\mu, \sigma^2)$. This is the variational part of the VAE, as the parameters are optimized with Variational Bayes. An encoder is trained to estimate parameters μ and σ, and a decoder to reconstruct images from latent variables sampled from the approximate posterior. This training process is guided by optimising the evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{h})\right] - \mathrm{KL}(q_\phi(\boldsymbol{h}|\boldsymbol{x})||p_\theta(\boldsymbol{h})).$$

Here the first term is the reconstruction loss which with the VAE assumptions simplifies to regular MSE loss as for a regular AE. The second term can be viewed as a regularization term, and is the Kullback–Leibler divergence (KLD), a measure of the distance between two probability distributions.

To actually backpropagate through the network and through the sampling operation, we require the reparametrization trick. Instead of sampling directly from the latent distribution, a random variable $\boldsymbol{\epsilon}$ is introduced to take care of randomness outside of the network, such that

$$\boldsymbol{h} = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + \boldsymbol{\sigma}_\phi(\boldsymbol{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ are the mean and variance produced by the encoder network and $\odot$ represents an element-wise product. This way the entire process becomes differentiable, making it possible to use standard gradient based optimization such as Adam. Additionally, for two Gaussian distributions the KLD becomes,

$$\mathrm{KL}(\mathcal{N}(\mu,\sigma)||\mathcal{N}(0,1)) = -\frac{1}{2}(1+\log(\sigma^2)-\mu^2-\sigma^2),$$

further simplifying the implementation. Adapting this to the SISR task, as for other models, simply entails giving LR samples as input and using HR samples as the loss target.

VAEs are an incredible development, and now play a key role in modern methods such as stable diffusion [31] and hybrid architectures like VAE-GANs [32]. Although powerful, they are known to produce blurry images compared to for example GANs, which was the state of the art image generation method at the time the first VAE paper was published. Here, the most successful REDNet model is used as an AE starting point and developed into a VAE. The main aim of this probabilistic implementation of SISR is to explore what kind of variance the model exhibits in its HR outputs.

5. *Generative Adversarial Network*

GANs have been incredibly successful in computer vision tasks such as image-to-image translation, image inpainting, and image super-resolution. Compared to the often blurry output from VAEs, GANs are know to produce crisp, realistic images. They were considered the state of the art shortly after being introduced by Goodfellow *et al.* [33] in 2014, maintaining their dominance until recently, when diffusion models began to take off.

GANs are based on a game-theoretic framework in which a generator network competes against an adversarial discriminator. The generator is in our case tasked with producing HR samples , while the discriminator attempts to assess whether a given HR sample is real (from the training data) or fake (produced by the generator), outputting a probability score indicating its judgment. In practice, this means that two networks are trained simultaneously. To begin with the generator produces poor outputs and the discriminator does a bad job of distinguishing real and generated samples. As they interact with each other, both networks get better, with the generator trying to fool the discriminator who is getting better at detecting fakes.

In and attempt to leverage the high-frequency detail promoted by GANs, a VAE-GANs model is implemented to further improve HR output quality produced by the pure VAE model. The REDNet inspired VAE is implemented as a generator along with the discriminator proposed in the SRGAN model by Ledig *et al.* [34].

The classical GAN framework [33] formulates a minimax game between a generator G and a discriminator D via the value function

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{y}\sim p_{\text{data}}}[\log D(\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{h}\sim p(\boldsymbol{h})}[\log(1-D(G(\boldsymbol{h})))].$$

In this VAE-GAN model, the generator maps a LR input $\boldsymbol{x}$ to a latent distribution $q_\phi(\boldsymbol{h}|\boldsymbol{x})$ and generates a HR output $\hat{\boldsymbol{y}}$. Its loss combines the previously defined reconstruction and KL divergence terms, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{KL}}$, with an adversarial loss term:

$$\mathcal{L}_G = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} - \lambda_{\text{adv}}\log D(\hat{\boldsymbol{y}}).$$

The discriminator is trained to distinguish real from generated samples by maximizing binary cross-entropy loss

$$\mathcal{L}_D = \mathbb{E}_{\boldsymbol{y}}[\log D(\boldsymbol{y})] + \mathbb{E}_{\hat{\boldsymbol{y}}}[\log(1-D(\hat{\boldsymbol{y}}))].$$

This combined objective leverages the latent structure learned by the VAE and the realism enforced by adversarial training.

D. Performance Metrics

1. *Peak Signal-to-Noise Ratio*

The pixel wise MSE between a ground truth HR sample $\boldsymbol{y} \in \mathbb{R}^{H\times W}$ and LR based model output $\hat{\boldsymbol{y}} \in \mathbb{R}^{H\times W}$ is given as:

$$\mathrm{MSE} = \frac{1}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\left(y[i,j]-\hat{y}[i,j]\right)^2,$$

but may not always be indicative of the actual perceived similarity of two images. The most widely used image quality assessment metric in the SISR field is the Peak Signal-to-Noise Ratio (PSNR):

$$\mathrm{PSNR} = 10\cdot\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right),$$

where MAX is the maximum possible pixel value of an image, 255 for RGB images. Since we are not working with standard images but rather with brightness temperature data, PSNR is computed separately for the two polarization channels, using the maximum value observed in the training data for each polarization.

2. *Structural Similarity Index Measure*

Unlike pixel-wise metrics such as the MSE and PSNR, Structural Similarity Index Measure (SSIM) is a perceptual metric designed to align with human visual interpretation by capturing structural distortions.

$$\mathrm{SSIM} = \frac{2\mu_y\mu_{\hat{y}} + c_1}{\mu_y^2\ \mu_{\hat{y}}^2 + c_1} \cdot \frac{\sigma_{y\hat{y}} + c_2}{\sigma_y^2\ \sigma_{\hat{y}}^2 + c_2}.$$

Here, μ_y and σ_y^2 are the mean and variance of $\boldsymbol{y}$, $\sigma_{y\hat{y}}$ is the covariance between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, and c_1 and c_2 are constant relaxation terms.

E. Dataset

The dataset is made up of temporally collocated passive microwave satellite retrievals from the SSM/I and AMSR sensors. These are daily retrievals spanning 2003 to 2020. Table I shows all relevant sensors and which part of the data timeline they cover. It should be noted that there is a small gap in the AMSR data from November 2011 until July 2012 due to AMSR-E reaching its end of life before it was replaced by AMSR2. There are a total of 6206 samples over the 18 year period which have been split into training, validation and testing subsets. Years 2003 - 2018 make up the training set with 5477 samples, and years 2019 and 2020 are used as validation and testing sets with 365 and 364 samples, respectively. Every sample contains both AMSR (HR) and SSM/I (LR) data. Each of these has 416 × 416 pixels and two channels, vertical and horizontal polarization. The LR satellite retrievals have a spatial resolution of about 25 km, whilst HR retrievals have twice the resolution at 12.5 km.

Each sample contains measurements over the ocean, while missing values over land, denoted by NaNs, are replaced with zeros. The data is normalized by subtracting the mean and dividing by the standard deviation computed from the training data.

TABLE I. Overview of sensors used for all passive microwave retrievals in the dataset.

Sensor	Start-date	End-date	Resolution
SSMI-F15	2003-01-01	2005-12-31	25 km
SSMIS-F16	2006-01-01	2008-03-31	25 km
SSMIS-F17	2008-04-01	2010-03-31	25 km
SSMIS-F18	2010-04-01	2020-12-31	25 km
AMSR-E	2003-01-01	2011-10-31	12.5 km
AMSR2	2012-07-01	2020-12-31	12.5 km

F. Implementation, Training and Hardware

All and models are implemented using PyTorch [35]. Model codes, data processing and figures can be found on GitHub at: `https://github.com/Alessimc/FYS5429`. Codes all either use the original implementations with some alterations or are reimplemented based on the described architecture. Training is conducted on an NVIDIA A100 80GB GPU, with a set maximum runtime of 24 hours and an early stopping criterion of no improvement over 20 epochs. To accelerate computations and reduce memory use, training is implemented using Automatic Mixed Precision. This allows some operations to be executed using lower precision floating point datatypes to speed up training on NVIDIA GPUs with Tensor Cores.

III. RESULTS AND DISCUSSION

A. Deterministic Models

Starting with the default SRCNN, Figure 2 shows a clear tendency of smaller batch sizes converging towards lower validation MSE values. We observe a slight increase in noise over the training period, but considering the performance gain, the stability of the training with batch size eight is deemed satisfactory. As we are most interested in comparing the SISR skill of varying architectures, all subsequent models are also trained using the same batch size. It should therefore be noted that this tuning on the SRCNN model may give it an unfair advantage, and other architectures may be able to achieve better results than presented here. However, this small batch size also comes at a lower computational cost, facilitating training of larger models that may encounter memory limitations.

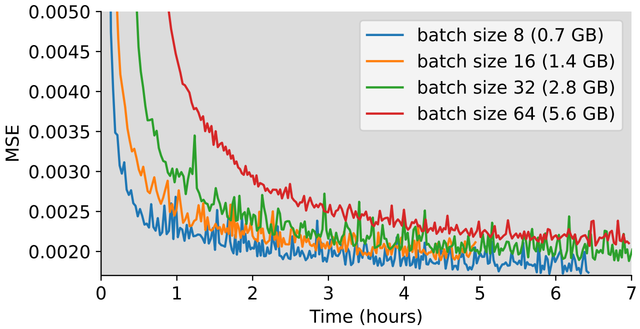

FIG. 2. SRCNN validation MSE over time for batch sizes 8, 16, 32 and 64. The legend shows memory usage in parentheses.

Figure 3 shows the comparison of four SRCNN models with varying depth and width considering validation MSE. Here SRCNN_L4 expands the default three-layer SRCNN to four layers, SRCNN_W128 and SRCNN_W256 maintain the same three layers, but expand the original 64 channels to 128 and 256, respectively. Increased depth shows little to no performance gain, whilst there is a clear advantage to having a wider model. These effects align well with equivalent results by Dong *et al.* [18] using typical RGB images as opposed to passive microwave satellite data. Any further mention of the SR-

CNN model refers to the best in terms of validation MSE SRCNN_W256 implementation.

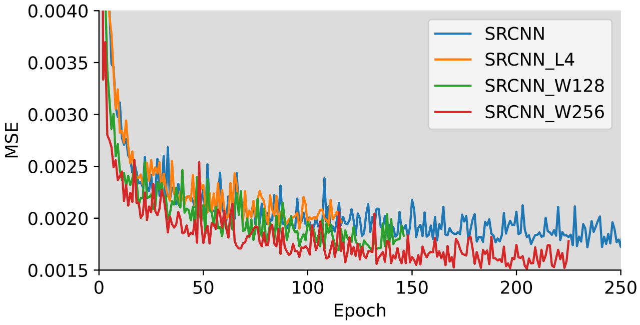

FIG. 3. SRCNN validation MSE for different model architectures. SRCNN is the default 3 layer model, SRCNN_L4 is a four layer version, SRCNN_W128 maintians the original 3 layers, but widens the channels from 64 to 128. Lastly, SRCNN_W256, widens the original model to 256 channels.

A similar analysis of the REDNet model is shown in Figure 4. Here we see the validation MSE for the three increasingly deep architectures proposed by Mao *et al.* [22] in addition to REDNet256; an implementation of REDNet10 with 256 channels as opposed to the original 128. As for the SRCNN, the AE shows little improvement with increased depth. Increasing the width of the shallowest REDNet10 model results in slightly noisier updates, but again lower validation MSE values. This increased width is therefore again chosen as the best model architecture and further mentions of REDNet refer to REDNet256.

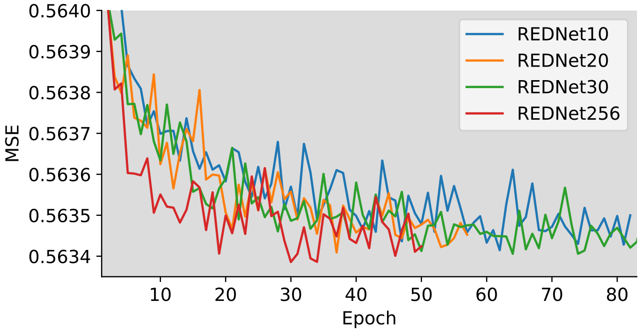

FIG. 4. REDNet validation MSE for different model architectures. REDNet10, -20 and -30 have 5, 10 and 15 convolutional–deconvolutional layer pairs, respectively. REDNet256, widens the REDNet10 model from 128 to 256 channels.

Notably, the MSE values for REDNet are two orders of magnitude larger than seen for the SRCNN. This is not because REDNet is doing a much worse job at the SISR task. As we can see in the right panel of Figure 5, REDNet reproduces values in the ice and ocean domains very well, producing a similar output to the HR ground truth to the left. However, a complete mismatch is observed in all continental regions, or more importantly, all nan-valued regions. To allow the ML models to handle these regions, all NaN values are set to zero as seen in the HR ground truth. Compared to all other models presented here, REDNet learns to zero out these regions already in the scaled training space which become 126 (the mean value used when scaling the data) when denormalized as seen in green in the right panel of Figure 5. During training, this zero value does not align with the expected value in these regions and leads to the high validation MSE.

Although it is not clear why this happens, it shows a clear difference in how models handle large outlying values, emphasising the importance of how NaN values are handled. Since the continental domain is not included in the assessment of SISR skill, this does not affect REDNet performance compared to other models. A masked loss function ignoring land was also tested, giving similar values as seen for the SRCNN, but having no noticeable effect on final SISR performance. For the sake of consistency across the training of different models, the regular loss function was therefore kept.

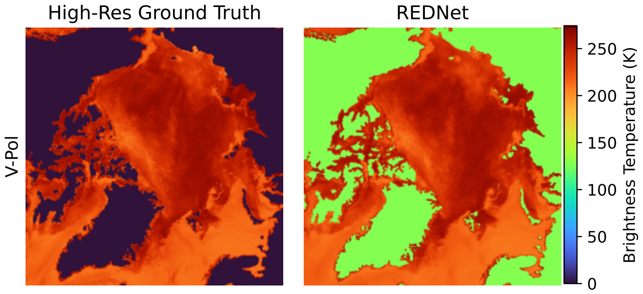

FIG. 5. Brightness temperatures over the Arctic Ocean and sea ice on February 16th 2020. The left panel shows the HR ground truth, whilst the right shows the REDNet generated output.

Both the SRCNN and REDNet show increased performance with width, but not depth. This may indicate that the passive microwave data has limited hierarchical complexity, such that the SR task does not benefit significantly from multiple layers of abstraction. Instead, the relevant features for resolution enhancement, such as edge sharpness or local gradients, may be relatively shallow and spatially localized. In this case, wider layers, which offer a greater number of channels to capture diverse low-level patterns, are more useful than deeper architectures that attempt to extract higher-level representations.

Additionally, the dataset may not contain enough samples or variety to effectively train deeper networks, increasing the risk of overfitting. Deeper models typically require more data and regularization to generalize well, and in the absence of this, their additional capacity can be wasted or even detrimental. These observations suggest that for this data, architectural improvements should focus more on increasing representational capacity per layer rather than stacking more layers. Here, we only explore model widths up to a maximum of 256 channels, which in both cases resulted in the best-performing configurations. Further research should be conducted to assess whether this performance gain continues for even wider architectures.

Although SwinIR and, even more so, ELAN employ relatively memory-efficient attention mechanisms compared to earlier transformer-based architectures, the primary challenge in training these models was the 80 GB GPU memory limit. Moreover, since this GPU is shared among multiple users, the full memory capacity is rarely available in practice. Training the full ELAN model as proposed by Zhang *et al.* [29] proved infeasible for full-resolution 416 × 416 input samples due to excessive memory requirement. Even when downsampling the inputs to 208 × 208 by selecting every second pixel, the model required 74.6 GB of memory, approaching the GPU's limit.

As a result, simplified versions of both attention-based models were implemented, significantly reducing memory demands while retaining their core architectural principles (see section II C 3 for details). These adaptations allowed for full-resolution training but may have constrained the models' representational capacity and ultimate performance. As shown in Figure 6, even in their simplified forms, both SwinIR and ELAN required significantly more memory and training time compared to the other models. Notably, ELAN did not meet the early stopping criterion and continued training until the 24-hour time limit. SwinIR also approached this limit, completing its training in just over 23.5 hours. In contrast, SRCNN and REDNet completed training in approximately eight and four and a half hours, respectively. In terms of memory consumption, both attention-based models demanded around 40 GB, whereas the other models required only about two GB. Thus, these attention-based models required at least about three times the training time and 20 times the memory compared to the other two models.

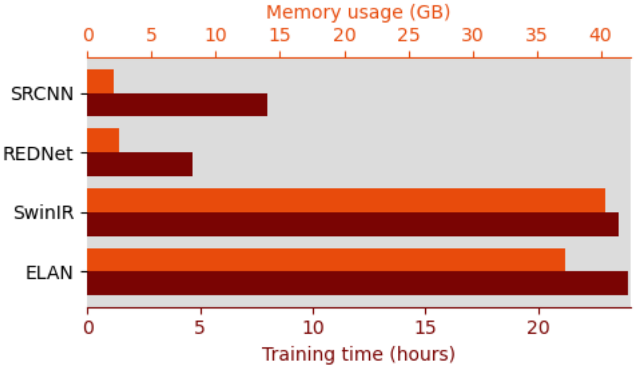

FIG. 6. Comparison of training time and memory usage for all deterministic models. Orange denotes the memory usage in GB whilst dark red is the training time in hours. Note that ELAN training stops at the maximum training time of 24 hours.

Although these attention based models show some skill in SISR of passive microwave satellite retrievals, with the available computational resources they are clearly outperformed by the much lighter REDNet model, as seen in Table II. These methods are therefore not explored any further in this study, but should be revisited in future work with greater GPU memory availability.

TABLE II. PSNR and SSIM metrics presented as a mean over the test set for vertical and horizontal polarizations. Shown is a comparison of all deterministic models.

	PSNR		SSIM	
	V-Pol	H-Pol	V-Pol	H-Pol
Baseline	33.7	31.2	0.791	0.769
SRCNN	38.0	33.1	0.830	0.820
REDNet	**39.3**	**34.8**	**0.867**	**0.854**
ELAN	38.2	33.3	0.835	0.825
SwinIR	37.6	33.1	0.835	0.824

Table II summarizes the performance of the four deterministic SISR models with mean PSNR and SSIM values over the entire test set. REDNet performs best across all metrics and polarizations, with a PSNR of 39.3 for vertical polarization (34.8 for horizontal), compared to the baseline value of 33.7 (31.2), and an SSIM of 0.867 (0.854), outperforming the baseline of 0.791 (0.769). However, all models achieve a notable improvement compared to the baseline value, with a PSNR improvement of at least 4 (2) and SSIM improvement of at least 0.039 (0.051).

These are quantitative metrics are typically used for RGB images and their exact applicability to SR of passive microwave data is uncertain. Even in traditional SISR studies, some models produce impressive metrics, but comparatively poor qualitative image results. Figure 7 shows example outputs from each model along with the LR input and HR ground truth. Close inspection clearly reveals that all models are truly able to improve the perceived resolution compared to the LR input, indicating that the PSNR and SSIM are reasonable metrics.

Focusing on an smaller domain, Figure 8 shows a small region including Svalbard and the east coast of Greenland. Here we observe a clear boundary between higher and lower brightness temperature values. This is the ice edge, the transition from open ocean in the south to sea ice pack in the north. Focusing on this structured boundary it is even clearer that all models improve perceived resolution. The keen observer will also be able to identify a few more structural features in the REDNet output compared to other models, aligning well with the quantitative metrics in Table II.

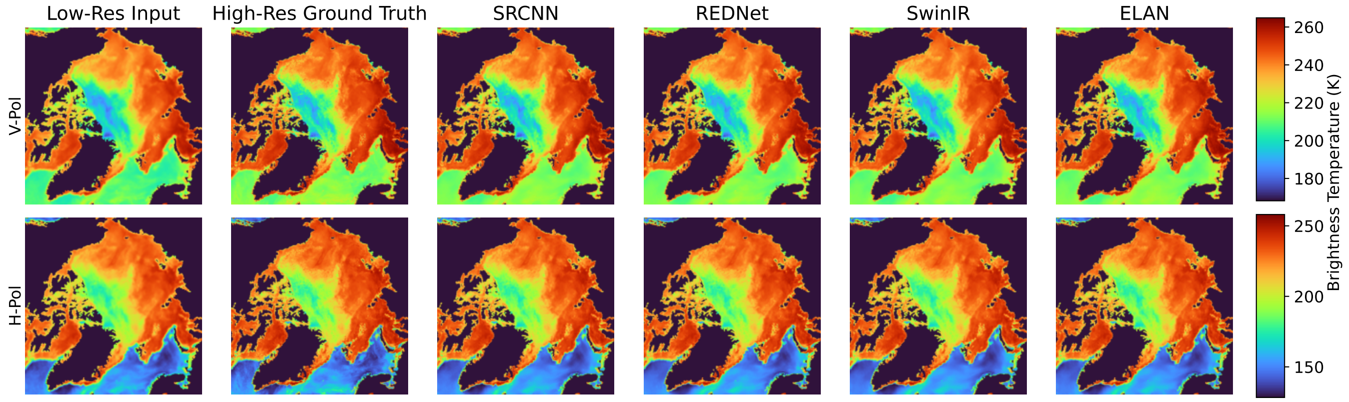

FIG. 7. Low- and high resolution brightness temperatures over the Arctic on February 16th 2020 are shown as well as super-resolved outputs from all deterministic models. Vertical polarization retrievals are in the upper row and horizontal retrievals in the lower row. Due to limited image size presented here, the reader is invited to inspect the full figure available in the GitHub repository: https://github.com/Alessimc/FYS5429/blob/main/final_figures/compare_models_full_domain.pdf

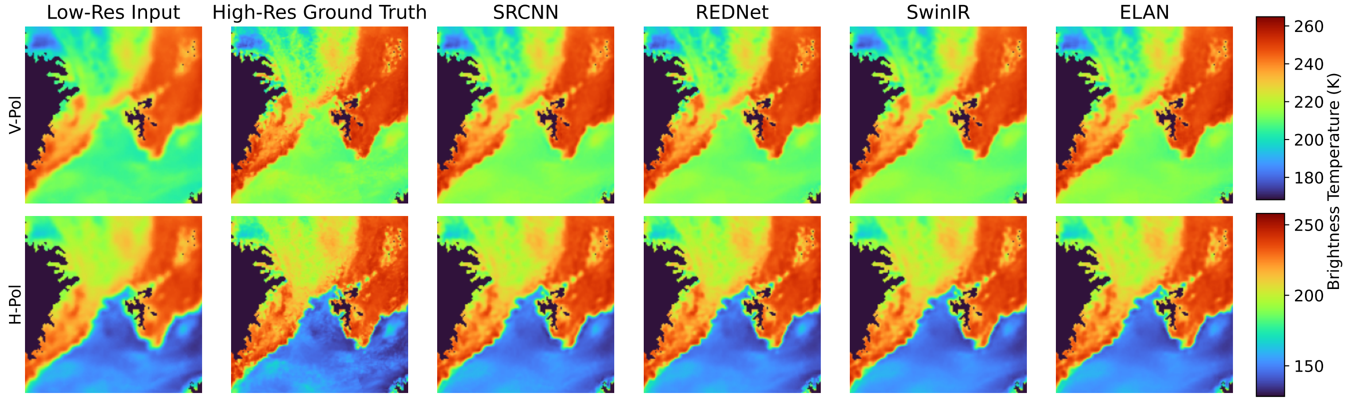

FIG. 8. Low- and high resolution brightness temperatures in a zoomed domain of Figure 7 including Svalbard and the east coast of Greenland on the February 16th 2020 are shown as well as super-resolved outputs from all deterministic models. Vertical polarization retrievals are in the upper row and horizontal retrievals in the lower row.

B. Probabilistic and Adversarial Model

Although the models presented so far perform relatively well at the SISR task, their deterministic nature is a limitation. The HR outputs look better than the LR input, but without a HR ground truth for comparison, it is impossible to say whether this is a physically plausible representation. This same limitation applies to a single output from a probabilistic model, but by comparing several of the generated HR scenes from a single LR input, it may be possible to understand in what regions the model is most uncertain.

The success of the REDNet model motivates its use as a starting point for the implementation of models capable of probabilistic HR outputs. We achieve this by developing it into a VAE. Since VAEs are known to produce blurry outputs, a VAE-GAN model is also implemented as GAN models were considered state of the art for computer vision tasks over several years, producing highly realistic high resolution images.

During initial training, the VAE exhibited signs of posterior collapse, evidenced by the KLD term approaching zero and the loss of probabilistic output ability. This issue may have stemmed from numerical instability, as it was resolved after code refactoring. Training was further stabilized by introducing KL annealing: gradually increasing the KLD term's weight from 0 to 1 during training.

The GAN-VAE model suffered from unstable training, with signs of divergence during optimization. To address this, several stabilization techniques were attempted, including label smoothing, gradient clipping, and assigning a lower learning rate to the discriminator. These methods aimed to reduce training instability by mitigating sharp gradients, encouraging smoother convergence, and preventing the discriminator from overpowering the generator, but had little to no effect.

Although GANs are capable of producing high-quality outputs, they are notoriously difficult to train due to the simultaneous optimization of two competing networks. Achieving stable and effective training often requires extensive hyperparameter tuning and architectural adjustments. Future work should consider applying already successful SR models such as SRGAN [34] to passive microwave data and subsequently explore new combinations such as the proposed REDNet based VAE-GAN.

Mean test PSNR and SSIM results are shown in Table III. The pure REDNet model still performs best, although the VAE only suffers minor degradation, whilst having the advantage of being able to produce probabilistic outputs. The VAE-GAN model on the other hand performs poorly, achieving both PSNR and SSIM scores below the baseline values. This is as expected due to the diverging training.

TABLE III. PSNR and SSIM metrics presented as a mean over the test set for vertical and horizontal polarizations. Comparing the overall best REDNet with a VAE and VAE-GAN version of the model.

	PSNR		SSIM	
	V-Pol	H-Pol	V-Pol	H-Pol
Baseline	33.7	31.2	0.791	0.769
REDNet	**39.3**	**34.8**	**0.867**	**0.854**
VAE	39.0	34.5	0.857	0.846
VAE-GAN	32.6	30.32	0.713	0.741

In Figure 9 we clearly see that the VAE-GAN model produces something resembling the LR input shown in Figure 7, even introducing some additional smoothing. The VAE on the other hand does not suffer from the expected blurriness and produces a very similar output to the REDNet in Figure 7. Computing the variance over 20 VAE generations using the same LR input results in Figure 10. Overall we observe low values, however there is a tendency for higher variance along the ice-ocean boundary and in regions containing more structure in Figure 9. There is some more widespread variance for the horizontal polarization and both have a patch of high variance, denoted by a red patch in the centre of the image. This coincides with regions where we would expect older sea ice to reside. Such older ice, typically has a greater surface roughness [36] which affects passive microwave retrievals, an therefore possibly the model.

Both the lack of VAE smoothing and low variance may be attributed to skip connections. These directly transfer information from the input to the output, bypassing the latent space. This may be enabling higher resolution features to be preserved as well as giving the network the possibility to somewhat ignore the latent space. Alternatively, the low variance observed in VAE generated HR data may still indicate a tendency towards posterior collapse or possibly a high certainty of the HR state. Further research is required to fully asses the implications of this low output variance, and the architecture should be tested without skip connections.

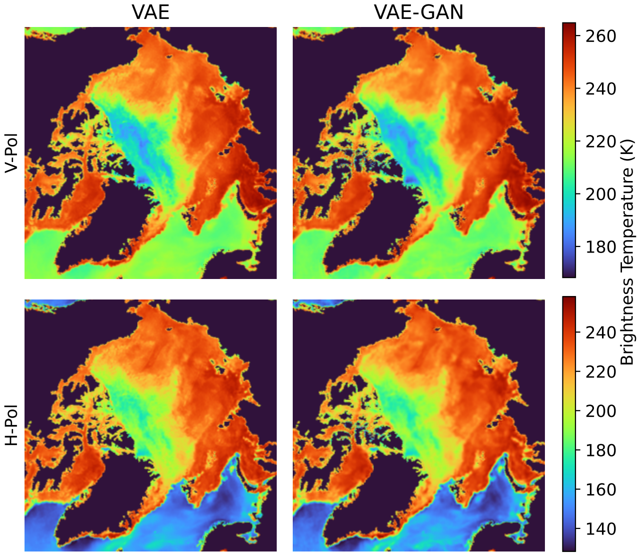

FIG. 9. SISR outputs from the REDNet based VAE (left)and VAE-GAN (right) model for February 16th 2020. Vertical polarization data is in the upper row and horizontal polarization in the lower row.

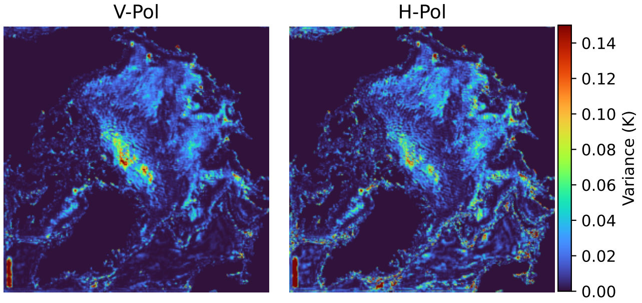

FIG. 10. Shown is the variance of the VAE generated outputs for February 16th 2020 over 20 different generations. Vertical and horizontal polarization results are shown in the left and right panels, respectively.

C. Seasonal Dependence

So far models have been assessed based on their mean quantitative performance over the entire test set and specific qualitative performance considering the case of February 16th 2020. As both sea ice and atmospheric conditions vary substantially throughout the year, it is reasonable to suspect some seasonal signatures in the model performance due to changes in the passive microwave emissions. In Figure 11 we see PSNR improvement for each day in the year long test dataset. Here, PSNR improvement is defined as

$$\text{PSNR improvement} = \text{PSNR}_{\text{model}} - \text{PSNR}_{\text{baseline}},$$

and all models that achieve and improvement over the baseline value are included.

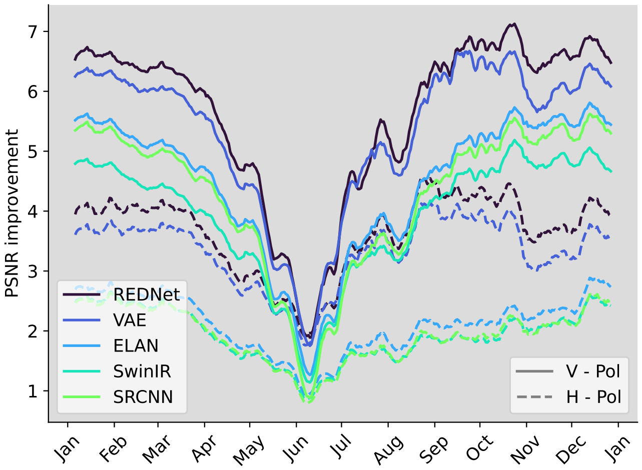

FIG. 11. PSNR metric improvement relative to the baseline for each day in 2020 (test dataset). All models outperforming the baseline are shown. Solid and dashed lines represent vertical and horizontal polarization performance, respectively. The data is smoothed using a 10-day rolling mean.

As indicated by the mean test metrics (see Table II and Table III), and further illustrated by the clustering of higher solid lines (vertical polarization) and lower dashed lines (horizontal polarization) in Figure 11, all models consistently perform better on vertical polarization data. This discrepancy may be attributed to the typically higher variability in horizontal polarization values, which are more sensitive to atmospheric disturbances [37]. As a result, vertical polarization features occur within a narrower brightness temperature range and may be easier for the models to learn. However, this explanation has not been explicitly verified and should be regarded as speculative.

Despite the 10-day rolling mean applied in Figure 11, notable day-to-day and month-to-month variability remains, suggesting sensitivity to seasonal conditions or changes in surface and atmospheric properties. Most striking is the clear collapse of all models around June.

Sea ice algorithms are known to struggle with performance during the melt period [6]. The model collapse indicates that these SISR models may be encountering similar challenges as SR skill drops off drastically in May, with a minimum in June followed by a rapid increase in July. This coincides well with the peak melt period due to 24 hour solar radiation input. During this period, sea ice dielectric properties and thus radiative emissions change drastically with the increase of liquid water, masking radiometric differences between various ice types and open water. Furthermore, increases in atmospheric moisture during summer months [38] also influences passive microwave retrievals [39], further complicating accurate discrimination between surface types and features.

IV. CONCLUSIONS AND FUTURE WORK

In this study, we explored and evaluated a range of deep learning models to enhance the resolution of passive microwave satellite retrievals by performing SISR. The tested architectures include a CNN, an AE, a VAE, a GAN, and two transformer-inspired attention-based models.

Among these, the autoencoder REDNet achieved the best performance, with a mean test PSNR of 39.3 for vertical polarization (34.8 for horizontal), compared to the baseline value of 33.7 (31.2), and an SSIM of 0.867 (0.854), relative to the baseline of 0.791 (0.769). These quantitative improvements correspond well with the perceptual increase in resolution. While attention-based models show promise, their potential is currently limited by significantly longer training times and greater cost, requiring about 20 times more memory and about three times the training time. Both SRCNN and REDNet showed minimal benefit from increased network depth but improved performance with greater model width. This motivates the study of even wider models than the 256 channels presented here.

Extending REDNet to a VAE architecture enabled probabilistic outputs at a slight cost in reconstruction accuracy, with a mean test PSNR of 39.0 (34.8) and SSIM of 0.857 (0.846). Over 20 HR outputs based on the same LR input, the VAE exhibited low output variance on the order of 10^{-1}, with the highest variability concentrated in areas with fine-scale spatial structure.

All models showed a marked collapse in SR skill during the melt period in June, indicating strong sensitivity to environmental conditions and the stability of satellite retrievals.

Since this work focused primarily on architectural comparisons, hyperparameter tuning was kept intentionally limited. Future work should include more comprehensive optimization for the most promising models, exploring broader parameter spaces including alternative loss functions and training schedulers. While minimizing MSE is known to improve PSNR, incorporating perceptual metrics such as PSNR and SSIM directly into the loss function should also be explored.

Some dataset-specific characteristics also warrant further investigation. The current dataset was preprocessed using a land mask, resulting in many NaN values that were converted to zeros. These skew the pixel distribution, possibly influencing model behaviour. Future work should investigate the use of unmasked datasets including retrievals over land to assess whether this influences performance.

In conclusion, this study demonstrates the clear potential of SISR methods for enhancing passive microwave satellite retrievals, with several model architectures yielding promising results. While the findings are informative, they also raise new questions, motivating further investigation into both the models presented here and the broader landscape of SR techniques and passive microwave datasets.

[1] T. Lavergne, A. M. Sørensen, S. Kern, R. Tonboe, D. Notz, S. Aaboe, L. Bell, G. Dybkjær, S. Eastwood, C. Gabarro, *et al.*, The Cryosphere **13**, 49 (2019).

[2] Y. Batrak and M. Müller, Geophysical Research Letters **45**, 6702 (2018).

[3] M. Müller, Y. Batrak, F. Dinessen, R. Grote, and K. Wang, Weather and Forecasting **38**, 1157 (2023).

[4] M. Rantanen, A. Y. Karpechko, A. Lipponen, K. Nordling, O. Hyvärinen, K. Ruosteenoja, T. Vihma, and A. Laaksonen, Communications earth & environment **3**, 168 (2022).

[5] J. Gower, *Oceanography from space*, Vol. 13 (Springer Science & Business Media, 2013).

[6] G. Spreen and S. Kern, in en*Sea Ice*, edited by D. N. Thomas (John Wiley & Sons, Chichester, UK ; Hoboken, NJ, 2017) third edition ed., pp. 239–260.

[7] S. C. Park, M. K. Park, and M. G. Kang, IEEE Signal Processing Magazine **20**, 21 (2003).

[8] R. Keys, IEEE Transactions on Acoustics, Speech, and Signal Processing **29**, 1153 (1981).

[9] C. E. Duchon, Journal of Applied Meteorology (1962-1982) , 1016 (1979).

[10] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, IEEE Transactions on Image Processing **18**, 969 (2009).

[11] J. Sun, Z. Xu, and H.-Y. Shum, in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008) pp. 1–8.

[12] S. Raschka, Y. H. Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python* (Packt Publishing Ltd, 2022).

[13] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, in *Advances in Neural Information Processing Systems*, Vol. 2, edited by D. Touretzky (Morgan-Kaufmann, 1989).

[14] K. He, X. Zhang, S. Ren, and J. Sun, IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, 1904 (2015).

[15] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, Z. Zhu, R. Wang, C.-C. Loy, X. Wang, and X. Tang, Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection (2014), arXiv:1409.3505 [cs.CV].

[16] Y. Sun, Y. Chen, X. Wang, and X. Tang, Advances in neural information processing systems **27** (2014).

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) `http://www.deeplearningbook.org`.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 295 (2016).

[19] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), arXiv:1412.6980 [cs.LG].

[20] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, Learning internal representations by error propagation (1985).

[21] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, in *Proceedings of the 25th international conference on Machine learning* (2008) pp. 1096–1103.

[22] X.-J. Mao, C. Shen, and Y.-B. Yang, arXiv preprint arXiv:1606.08921 (2016).

[23] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 770–778.

[24] D. Bahdanau, K. Cho, and Y. Bengio, CoRR **abs/1409.0473** (2014).

[25] A. Deviyani and U. Singh, .

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, arXiv preprint arXiv:2010.11929 (2020).

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, in *Proceedings of the IEEE/CVF international conference on computer vision* (2021) pp. 10012–10022.

[28] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, in *Proceedings of the IEEE/CVF international conference on computer vision* (2021) pp. 1833–1844.

[29] X. Zhang, H. Zeng, S. Guo, and L. Zhang, in *European conference on computer vision* (Springer, 2022) pp. 649–667.

[30] D. P. Kingma, M. Welling, *et al.*, Auto-encoding variational bayes (2013).

[31] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022) pp. 10684–10695.

[32] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, in *International conference on machine learning* (PMLR, 2016) pp. 1558–1566.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Advances in neural information processing systems **27** (2014).

[34] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 4681–4690.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, Pytorch: An imperative style, high-performance deep learning library (2019), arXiv:1912.01703 [cs.LG].

[36] M. Harder, Annals of Glaciology **25**, 237 (1997).

[37] S. Willmes, M. Nicolaus, and C. Haas, The Cryosphere **8**, 891 (2014).

[38] E. Jakobson and T. Vihma, International Journal of Climatology **30**, 2175 (2010).

[39] S. M. Løvås, I. Rubinstein, and C. Ulstad, Polar research **13**, 67 (1994).