Complex-valued Wasserstein GAN for SAR Image Generation

Victor Dhédin, Jérémie Levi, Jérémy Fix, Chengfang Ren, I. Hinostroza

COMPLEX-VALUED WASSERSTEIN GAN FOR SAR IMAGE GENERATION

*Victor Dhédin*** *Jérémie Levi*** *Jérémy Fix** *Chengfang Ren*† *Israel Hinostroza*†

** CentraleSupélec, Université Paris-Saclay, F-57000 Metz, France
* LORIA, CNRS, CentraleSupélec, Université Paris-Saclay, F-57000 Metz, France
† SONDRA, CentraleSupélec, Université Paris-Saclay, 91190 Gif-sur-Yvette, France

ABSTRACT

Complex-Valued (CV) Synthetic Aperture Radar (SAR) image generation and augmentation is an important pillar to enhance deep learning performance for SAR applications such as detection, classification, segmentation, super-resolution etc. Usual transformations (flip, rotation, translation, scaling, etc.) are mostly inapplicable to SAR images due to the radar characteristics and the processing pipeline. In this paper, we explore the applicability of Wasserstein Generative Adversarial Networks (WGANs) to SAR data. The latter, being complex, require CV generator and a discriminator taking as input a complex-valued signal, to capture the underlying distribution. In particular, we show the applicability of CV-WGANs for the synthesis of (i) Fourier spectrum of various MNIST like datasets as toy example and (ii) L-band UAVSAR dataset.

***Index Terms*—** Synthetic Aperture Radar, Complex Valued Neural Networks, Wasserstein Generative Adversarial Networks.

1. INTRODUCTION

Synthetic Aperture Radar (SAR) is a moving radar system using electromagnetic waves to sense Earth's surface. These systems are operational in almost all weather and illumination situations. Modern SAR systems can produce high-resolution Complex-Valued (CV) images with a sub-meter resolution. In stripmap SAR mode, many artifacts may appear depending on the acquisition setup: shadows (as in optical systems), foreshortening and layover (see [1, Chapter 2]). Therefore, SAR data cannot be augmented using classical transformations e.g. rotation, flip, scale, etc. An alternative approach is to use Generative Adversarial Networks (GANs) in which the generator should ultimately capture the underlying data distribution to lure the discriminator.

The original GAN [2] minimizes the Jensen-Shannon divergence between the real data distribution $\mathbb{P}_r$ and the generated data distribution $\mathbb{P}_\theta$ but is exposed to convergence difficulties with some common issues such as mode collapse (where the generated samples lack diversity) as well as difficulties in quantifying the quality of the generated samples. Then, the Wasserstein GAN (WGAN) [3] is proposed to be more resilient to these issues by minimizing the Wasserstein distance instead. However, these GANs [2–4] are not adapted to capture the distribution of CV data such as SAR or any data in Fourier representation. Even though CV features can be casted into 2D Real Valued (RV) ones, CV Neural Networks (CVNNs) [5, 6] have the advantage to naturally keep the connection between real and imaginary parts (phase information) which reduce the number trainable weights to accelerate the training, prevent overfitting and gain in stability compared to equivalent RV Neural Networks (RVNNs) [7–11].

In this paper, a new CV-Wasserstein GAN (CV-WGAN) is proposed with a CVNN discriminator extending the approach in [12]. In our case, the generator cannot lure the discriminator without correct phase information whereas the discriminator is only fed with amplitude in the previous work. The experiments show stable training phase without convergence and mode collapse issues and the generated images of MNIST and SAR are convincing. This paper is organized as follows. Section 2 introduces GAN and CVNN architectures, the training method and setups used for CV dataset. Section 3 analyzes the performance CV-WGAN in terms of learning and image synthesis for MNIST and FashionMNIST Fourier spectrum and SAR dataset following by concluding notes in section 4. Our inference codes are available online[1].

2. COMPLEX-VALUED WASSERSTEIN GAN

2.1. Background on WGANs and CVNNs

The setup of WGAN is similar to GANs by considering a generator $\mathcal{G}_\theta$, parameterized by θ, which maps a latent space $\mathcal{Z}$ to a data space $\mathcal{X}$ and a critic [2] $\mathcal{C}_w$ which maps the data space $\mathcal{X}$ to a scalar in $\mathbb{R}$. We denote $\mathbb{P}_z$ the probability distribution on the latent space $\mathcal{Z}$. The objective function is given by equation 1,

$$\mathcal{W}(\mathbb{P}_r, \mathbb{P}_\theta) = \mathbb{E}_{x\sim\mathbb{P}_r}[\mathcal{C}_w(x)] - \mathbb{E}_{z\sim\mathbb{P}_z}[\mathcal{C}_w(\mathcal{G}_\theta(z))], \quad (1)$$

which should be maximized by the critic and minimized by the generator and where the discriminator should be a $K-$Lipschitz function for some $K \in \mathbb{R}^+$. The optimization proceeds by alternating several optimization steps for

[1] https://anonymous.4open.science/r/complex-wgan-DE0B

[2] The second network is called a critic rather than a discriminator because it does not seek to perform binary classification anymore

the critic followed by one update of the generator. The authors suggested to clip the parameters to guarantee the existence of such a Lipschitz constant but the following work of [13] suggested to use a gradient penalty rather than a weight clipping to obtain improved quality and convergence of the WGAN. In this framework, a gradient penalty term $\lambda_{gp}(\|\nabla_w \mathcal{C}_w(\hat{x})\|_2 - 1)^2$, where $\hat{x}$ are a minibatch of interpolates between real data x and generated data $\tilde{x}$, is added to the loss minimized when optimizing the critic.

On the other hand, A CVNN is a map $\mathcal{F}_\omega$ parameterized by CV weights ω that predicts from inputs $\mathcal{X}_\mathbb{C}$ to outputs $\mathcal{Y}_\mathbb{C}$ where both spaces are complex. The network should follow complex product rule for fully connected layers and complex convolution for convolutional layers as defined in [14]. The activation functions are usually extended from RV ones according to [15]. The complex batch normalization forces hidden CV representations to be globally centered, unit variance and circular (null pseudo-covariance) as detailed in [14]. CV Weights initialization is similar to RV ones but keeping the independence between real and imaginary parts.

The main difficulty in training CVNNs is that the loss should remain RV and lower bounded which implies the training loss non holomorphic. In order to apply complex backpropagation, Wirtinger derivative is used as in [7, 14] which require 4 RV gradients or equivalently 2 CV gradients whereas 1 CV gradients is needed for holomorphic function $f(z)$ thanks to Cauchy-Riemann condition $\frac{\partial}{\partial z^*} f(z) = 0$. One can refer to [16, appendix B.6.2.] for the complex chain rule applied for feed forward neural networks. Once the CV gradients are computed, the weights can be updated using any optimizer strategy, e.g. ADAM, SGD, RMSprop etc.

In the following, we will use $\lambda_{gp} = 10$, with one generator update every $n_{critic} = 5$ updates of the discriminator. For both the update of the critic and the generator, the optimizer is Adam with a learning rate of $3e-5$, $\beta_1 = 0.5$, $\beta_2 = 0.9$.

2.2. Complex valued generators and discriminators

As explained in the previous sections, we consider CVNNs for the WGAN. Up to the knowledge of the authors, this setup has not been extensively studied. The work of [12] study the application conditional adversarial training Co-Vegan for the reconstruction of CV magnetic resonance images. They consider a complex valued U-Net generator and a RVNN critic. Note that since the critic is real valued, it does not base its decision on the phase while we will exploit the phase in our work for discriminating generated from real data. The loss they optimize is multi-objective, with, in addition to the WGAN loss, similarity losses between the output of the generator and the targeted reconstruction. The discriminator takes as input the modulus of the generated and real reconstructions, and all of the similarity losses but the content loss compares the generated reconstruction and the real reconstruction in the complex domain. In our work, we study a different problem where we focus on unconditional generation due to lack of labels in SAR imaging, and we consider a CV discriminator with an output in $\mathbb{R}$. Contrary to [12], the proposed discriminator can extract features from the real and fake data using both modulus and phase information.

In our experiments, we define two neural networks, a generator $\mathcal{G}_\theta : \mathcal{Z} \mapsto \mathbb{C}^n$ for synthesizing CV data and a critic: $\mathcal{C}_w : \mathbb{C}^n \mapsto \mathbb{R}$ where $w \in \mathbb{C}^{n_c}$ and $\theta \in \mathbb{C}^{n_g}$. In the rest of the paper, we consider two sets of architectures with the hyperparameters given in table 1 for the critic and table 2 for the generator. The architecture of the discriminator is a series of n^c_{disc} blocks `Conv` - $\mathbb{C}$`eLu` - `ConvDownsampling`-$\mathbb{C}$`eLu`. The convolutional layers have a kernel size of 3, the padding is 1 and stride 1 except for the downsampling where the padding is 0 and the stride 2. Downsampling is obtained with a strided convolution as a max downsampling is not possible with complex valued feature maps. The number of kernels of the first convolutional layer is 32 and is doubled at every downsampling layer but always upper bounded by n^*_c. The final layers are a global average pooling layer and a linear layer to compute one single scalar for every input. The final activation function, of the last linear layer, is the modulus.

	MNIST	SAR
Number of convolutional layers n^c_{disc}	5	8
Maximal number of channels n^*_c	128	512
Minimal spatial size s	8	4

Table 1. Critics setup for MNIST and SAR experiments

The architecture of the generator starts with a linear layer from the latent space n_z to a representation of size $c_0 \times s_g \times s_g$ which is reshaped as a 3d volume (c_0, s_g, s_g). Then comes a serie of n^c_{gen} blocks `Conv` - $\mathbb{C}$`eLu` - `ConvUpsampling`-$\mathbb{C}$`eLu`. The `ConvUpsampling` is an upsample using the nearest policy followed by a convolution. All the convolutions have a kernel size of 3, the padding is 1 and stride 1. The number of channels is multiplied by the channel multiplicator coefficient c_k at every upsample. The final layer is a convolution with a kernel size of 5, a stride 1 and a zero padding of 2. These architectures have been implemented using pytorch 1.12.1 and torchvision 0.13.1 [17].

	MNIST	SAR
Latent space dimension n_z	128	512
Number of convolutional layers n^c_{gen}	4	8
Number of filters c_0	64	512
Channel multiplicator c_k	1	0.8
Spatial dimension of the first layer s_g	4	2

Table 2. Generators setup for MNIST and SAR experiments

3. EXPERIMENTAL RESULTS

3.1. MNIST and FashionMNIST Fourier spectrum generation

We begin with toy problems synthesizing handwritten digits or FashionMNIST images in the Fourier space. The real data

are the MNIST digits or FashionMNIST item on which we apply a 2D-Fourier transform. As a reminder, these datasets comprise 60.000 samples as 28×28 grayscale images for the digits or clothings, the datasets being balanced. There are also 10.000 samples in a test fold which will be used to compute the validation metrics. To illustrate the applicability of CV-WGANs, the digits are transformed in the complex domain. The pixel values are divided by 255, the images are zero-padded to 32×32 and then transformed by computing their 2D Fourier transforms leading to CV representation. The Fourier spectrum is spatially shifted to center the zero frequency component. Once trained, to better visualize the performance of the generator, we compute the inverse Fourier transform on its output, which is certainly easier to qualitatively judge than its Fourier counterpart.

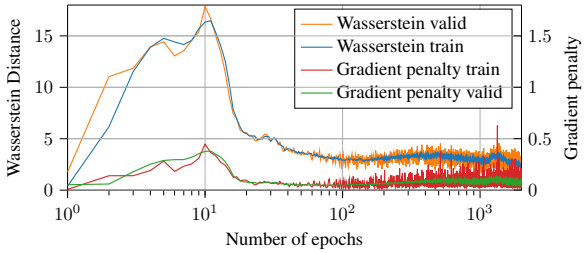

Fig. 1. The Wasserstein distance and gradient penalty terms during training on the Fourier MNIST dataset.

With the hyperparameters given in tables 1 and 2, the generator and discriminator have approximately 390.000 trainable parameters each. Training for 2000 epochs on an NVIDIA GeForce RTX 3090 takes approximately 23 hours with a batch size of 128.

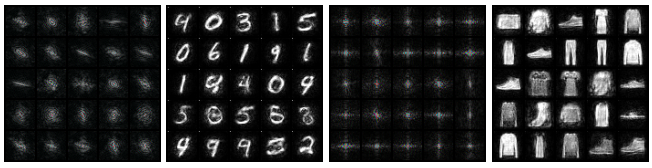

Fig. 2. From left to right: generated Fourier MNIST spectrums in HSV colorspace and the corresponding images by FFT inverse, generated Fourier FashionMNIST spectrums in HSV colorspace and the corresponding images by FFT inverse.

We ran the experiment for 2000 epochs. The Wasserstein distance and the gradient penalty terms are shown on figure 1. For all the runs we tried, the trend of the curves was the same : an initial low value, ramping up to some maximal value and then an almost steady decrease with a long tail. The asymptotic slope is usually very gentle. Also, the terms computed on the validation fold fluctuates significantly at the end of the run. Even if the computed Wasserstein distance is initially low, the samples generated during these very first epochs are of bad quality. Sample quality is usually correlated with a low Wasserstein distance only after some initial training of the networks, certainly required for the discriminator to extract meaningful features from the data. The metrics on both training and validation data are very close which probably indicate that there is no overfiting on the training set. We evaluated the Frechet Inception Distance [18] of the trained generators by computing the inverse Fourier transform, duplicating the grayscale channel and resizing the images as suggested in [19] to have 3 channels 299×299 images. For the Fourier MNIST dataset and generator, the FIDs are respectively 209 and 210 on the train and test folds. For the Fourier Fahsion MNIST dataset, the FIDs are respectively 170 and 171 on the train and test folds. These are similar but large values although, as we will see, the generated samples are qualitatively reasonably good. We believe this is related to the fact the inception V3 network used for computed the FID are trained on ImageNet with features that are probably not well calibrated for samples such as MNIST or Fourier MNIST.

We now study more qualitatively the structure of the latent space and the performances of the trained generator. Recall that the generator is trained to generate representations as close as possible to a 2D Fourier transform of a digit. We sampled a grid of 10×10 independent random latent vectors z_i and computed the output of the generator $\mathcal{G}(z_i)$. These outputs are displayed on the left of figure 2. These samples are shown in the HSV colorspace. For a complex number $\rho \exp(i\theta)$, the associated HSV is set to $(\theta, 0.5+\hat{\rho}/2, \hat{\rho})$ where $\hat{\rho} = \log(1+\rho)$ normalized over the whole image in $[0, 1]$.

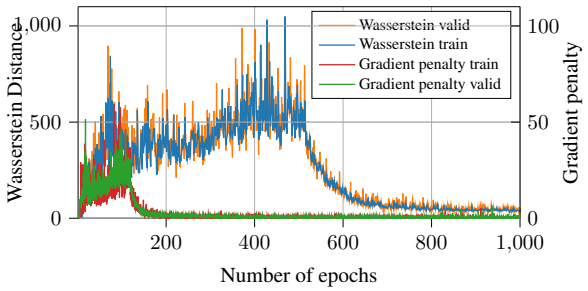

Fig. 3. The Wasserstein distance and gradient penalty terms during training on the SAR dataset.

3.2. Synthetic Aperture Radar image generation

For the application of WGANs on SAR data, we make use of the UAV-SAR data provided by the NASA JPL. We used the 2×8 SSurge 15305 with HH polarization, west Los Angeles (CA) for training and 2×8 SSurge 12600 with HH polarization, south east Los Angeles (CA), for validation. Configuration 2×8 for 3.3m $\times$ 4.8m pixel spacing (slant-range $\times$ azimuth). Both the training and validation SLC data were collected in L-band (1.26 GHz). The flight was on November 20^{th}, 2014. The training SLC stack of shape $(6719, 4893)$ is randomly cropped into thumbnails of size $(256, 256)$. Every complex point $z = \rho \exp(i\theta)$ has its norm transformed

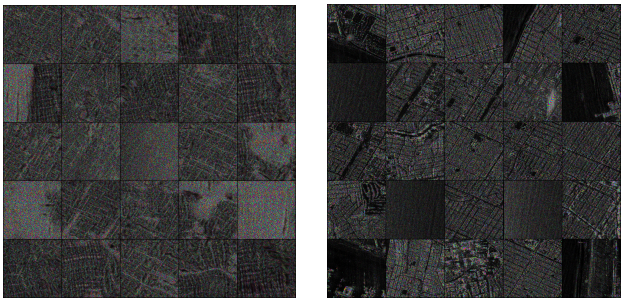

Fig. 4. Left: Generated samples by the generator from a grid of 5×5 independent latent vectors. The magnitude and the phase of the complex outputs are used to display images in the HSV colorspace. Right: Real SAR images provided for comparison with a variety of landscapes : urban, sea, sea side.

by applying $\log(\rho + 10^{-5}) \exp(i\theta)$ and then multiplied by 0.4. The latter is an empirically found normalizing factor. For these data, we define one epoch when 494 patches have been processed.

With the hyperparameters given in tables 1 and 2, the generator and discriminator have approximately respectively 8 million and 12 million trainable parameters. Training for 4400 epochs takes 24 hours on a single GPU of an NVIDIA A100-SXM4-40GB, with a batch size of 25. We ran the experiment for almost 4500 epochs. The Wasserstein distance and the gradient penalty terms are shown on figure 3. For all the runs we tried, the trend of the curves was the same and similar to the training on the Fourier spectrum of MNIST or FashionMNIST dataset. The Wasserstein distance and gradient penalty are similar on the training and validation folds although they are much more fluctuating on the SAR data than on the Fourier spectrum of MNIST or FashionMNIST data and this for both the validation and training data. Quantitatively, the FIDs are respectively 118 and 142 on the train and validation folds. We believe these values are relatively high for the same reason than for the toy datasets.

As for the previous experiments, we represent independently generated samples on figure 4. The independently generated samples show a diversity of patterns although the quality of the generated data are harder to judge than for the MNIST like databases. The two most recognizable patterns are the regular lines patterns which seem to be urban like landscapes and reminds the regular structure of the Los Angeles neighborhoods where the training data originates. The second pattern is the uniform bright pattern which is similar to SAR data over the sea (note the images are normalized to have a higher contrast in the paper). For easing the interpretability of these generated data, real SAR data are shown on the right of figure 4. We recognize the various patterns of urban zones, sea, sea side and also a mountain for the image on the bottom left of the real SAR images.

Quantitatively from the Wasserstein distance and qualitatively from the samples we presented, it seems the generator is able to generate CV SAR images. Although it is not always straightforward to appreciate the quality of the generated images, we do observe some patterns that remind the ones observed in real data. Also, we note that when patterns reminding streets are generated, these patterns appear as regular as in large US cities with the grid pattern. The patterns are not blurry: they do have high frequency contents such as the sharp edges of the streets and also some zones that appear to be of specific types (sea, sea side, city center). Certainly these details could be better appreciated if we could generate larger patches. The computational resources required for training the WGAN precluded the training for larger patches.

4. DISCUSSION

In this paper, we studied complex-valued generative adversarial networks and especially the Wasserstein variant. Our initial experiments with the original GAN formulation on the same data has confirmed the difficulties in training vanilla GANs. Training was easier with the WGAN formulation although it required a long investigation of different network architectures trying to balance the performance of the generator and the discriminator.

Given the difficulty of interpretation of the generated SAR data, we found it interesting to begin the investigation by training a CV-WGANs on the 2D Fourier spectrum of the famous MNIST and FashionMNIST datasets. These experiments revealed the ability of training CV discriminator and CV generator and then the quality of the synthesized samples. Various interpolation experiments revealed the continuity of the learned latent encoding. The same experiments carried out on the SAR data also demonstrated the ability to train a generator for synthesizing CV SAR images of size 256×256. Beyond the Wasserstein distance which steadily improves during training, the quality of the samples seems to be relatively good and these generated samples exhibit diversity with urban like or sea and sea side like patterns.

These promising results open other perspectives. As indicated in the introduction, data augmentation of SAR data can be tricky using standard approaches. The ability to generate fake but realistic SAR images allows to envision the application of GANs to data augmentation. The next step would be studying the training of conditional GANs to be used for data augmentation for supervised learning (e.g. classification or segmentation). Beyond data augmentation, conditional GANs have also demonstrated interesting performances in image translation (e.g. Pix2Pix [20]) and superresolution. These are also tasks of interest in the SAR community.

Acknowledgments

This work was performed using resources of the Mésocentre of CentraleSupélec, École Normale Supérieure Paris-Saclay and Université Paris-Saclay supported by CNRS and Région Île-de-France as well as resources of the Data Centre pour l'Education [21].

5. REFERENCES

[1] Africa Flores, K. Herndon, Rajesh Thapa, and Emil Cherrington, *The SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation*, 04 2019.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, Eds. 2014, vol. 27, Curran Associates, Inc.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds. 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 214–223, PMLR.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2016.

[5] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 541–551, 2012.

[6] A. Hirose, *Complex-valued Neural Networks*, vol. 400, Springer Science & Business Media, 2012.

[7] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez, "Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2990–2994.

[8] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez, "Comparison between equivalent architectures of complex-valued and real-valued neural networks-application on polarimetric sar image segmentation," *Journal of Signal Processing Systems*, vol. 95, no. 1, pp. 57–66, 2023.

[9] Y. Cao, Y. Wu, P. Zhang, W. Liang, and M. Li, "Pixel-wise PolSAR image classification via a novel complex-valued deep fully convolutional network," *Remote Sensing*, vol. 11, no. 22, pp. 2653, 2019.

[10] J. Zhao, M. Datcu, Z. Zhang, H. Xiong, and W. Yu, "Contrastive-Regulated CNN in the Complex Domain: A Method to Learn Physical Scattering Signatures from Flexible PolSAR Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10116–10135, 2019.

[11] Q. Sun, X. Li, L. Li, X. Liu, F. Liu, and L. Jiao, "Semi-supervised complex-valued GAN for polarimetric SAR image classification," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019)*. IEEE, 2019, pp. 3245–3248.

[12] B. Vasudeva, P. Deora, S. Bhattacharya, and P.M. Pradhan, "Compressed sensing mri reconstruction with co-vegan: Complex-valued generative adversarial network," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 1779–1788.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, NIPS'17, p. 5769–5779, Curran Associates Inc.

[14] C. Trabelsi, O. Bilaniuk, Y. Zhangh, D. Serdyuk, S. Subramanian, J.F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C.J. Pal, "Deep complex networks," in *International Conference on Learning Representations*, 2018.

[15] Y. Kuroe, M. Yoshid, and T. Mori, "On activation functions for complex-valued neural networks: existence of energy functions," in *Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP 2003*, pp. 985–992. Springer, 2003.

[16] J. A. Barrachina, *Complex-valued neural networks for radar applications*, thesis, Université Paris-Saclay, Dec. 2022.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, 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," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash

equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[19] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu, "On aliased resizing and surprising subtleties in gan evaluation," in *CVPR*, 2022.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[21] J. Fix, S. Vialle, R. Hellequin, C. Mercier, P. Mercier, and J.-B. Tavernier, "Feedback from a data center for education at centralesupélec engineering school," in *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2022, pp. 330–337.