Real- and Complex-Valued Neural Networks for SAR Image Segmentation Through Different Polarimetric Representations

Jose Barrachina, Chengfang Ren, Gilles Vieillard, Christele Morisseau, Jean-philippe Ovarlez

REAL- AND COMPLEX-VALUED NEURAL NETWORKS FOR SAR IMAGE SEGMENTATION THROUGH DIFFERENT POLARIMETRIC REPRESENTATIONS

J. A. Barrachina[⋆†] *C. Ren*[⋆] *G. Vieillard*[†] *C. Morisseau*[†] *J.-P. Ovarlez*[⋆†]

[⋆] SONDRA, CentraleSupélec, Université Paris-Saclay, 91192 Gif-sur-Yvette, France
[†] DEMR, ONERA, Université Paris-Saclay, 91120 Palaiseau, France

ABSTRACT

In this paper, we investigated the semantic segmentation of Polarimetric Synthetic Aperture Radar (PolSAR) using Complex-Valued Neural Network (CVNN). Although the use of the coherency matrix is ubiquitous as the input of CVNN [1–7], Pauli vector is also a relevant representation despite the noise. Two equivalent networks Complex-Valued Fully Convolutional Neural Network (CV-FCNN) and Real-Valued Fully Convolutional Neural Network (RV-FCNN), equivalence in terms of trainable parameters, are compared using both Pauli vector and the coherency matrix as the input feature. Experimentation on San Francisco dataset illustrated a better accuracy of CV-FCNN against it real-valued equivalent.

Index Terms— Polarimetric Synthetic Aperture Radar, Complex-Valued Neural Network, semantic segmentation, Pauli representation, coherency matrix.

1. INTRODUCTION

Deep learning techniques are becoming widely popular and have extended into PolSAR image classification [8,9]. In particular, numerous publications use CVNN as an alternative to conventional Real-Valued Neural Network (RVNN) for radar applications [10, 11] since radar data are generally complex-valued thanks to In Phase and Quadrature (I-Q) channels. One of the first works on PolSAR image classification using deep learning was implemented in [1] using a Complex-Valued MultiLayer Perceptron (CV-MLP). These results were also replicated by [12]. Numerous articles thereafter proposed a Complex-Valued Convolutional Neural Network (CV-CNN) for performing PolSAR classification [2–5].

Contrary to the optical image, multiple objects or classes are generally present in the scene acquired by airborne and spaceborne SAR sensors due to its large aperture. The latter may imply SAR image classification from CNN inadequate or impossible since the latter requires a fixed input size to classify each object. Additionally, the resolution of SAR image is generally different from X-band, L-band or P-band acquisition which also has an impact on the object size in the SAR image. To get rid of input size in the classification process, pixel-wise classification of SAR image, namely semantic segmentation, received greater interest during the last decade.

In the recent advance of neural networks, Complex-Valued Fully Convolutional Neural Network (CV-FCNN), which is based on U-net architecture [13], is particularly designed to segment SAR images. Indeed the *state-of-the-art* performance of such networks are obtained in [6] and [7].

To our best knowledge, all the work mentioned above used the coherency matrix as their network input representation regardless the PolSAR application. Even though the coherency matrix is a well-known attribute in unsupervised PolSAR image classification used for e.g. H-α, Wishart classifier, etc. It might not be adapted in the supervised learning framework for pixel-wise segmentation tasks as explained in section 2.

In this paper, we propose an alternative input representation known as the Pauli vector. We explain why CV-FCNNs may better profit from this input format. Then we compare the performance of CV-FCNN against an equivalent Real-Valued Fully Convolutional Neural Network (RV-FCNN) using either Pauli vector or coherency matrix as input features through the segmentation of San Francisco open-source PolSAR dataset.

Section 2 presents the dataset and its pre-processing techniques. Section 3 explains the model architectures used for the experiments. Finally, section 4 compares the performance of the proposed neural networks.

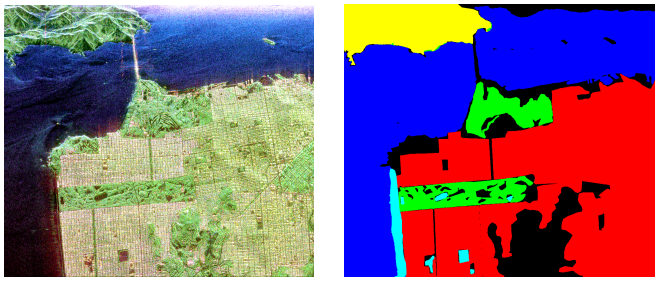
2. DATASET

PolSAR makes use of signal coherence (or equivalently phase and local phase variance) existing on any single look complex data channels S measured in the horizontal (H) and vertical (V) transmit/receive polarimetric channels known as the Sinclair scattering matrix:

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix}. \quad (1)$$

For each pixel of the Synthetic Aperture Radar (SAR) image, the backscattering vectors are usually expressed in

The authors would like to thank the Délégation Générale de l'Armement (DGA) for funding and the Metz campus of CentraleSupélec for providing the DCE cluster to run our simulations.



(a) SF-AIRSAR image (b) Ground truth

Fig. 1: SF-AIRSAR image and ground truth. **A** Mountain; **B** Water; **C** Urban; **D** Vegetation; **E** Bare Soil

Pauli basis and are vectorized onto one single complex vector $\boldsymbol{k} \in \mathbb{C}^3$ [14], so that

$$\boldsymbol{k} = -\frac{1}{\sqrt{2}}\left[S_{HH} + S_{VV}, S_{HH} - S_{VV}, 2\,S_{HV}\right]^T . \quad (2)$$

The Hermitian coherency matrix is formally built as:

$$\boldsymbol{T} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{k}_j\,\boldsymbol{k}_j^H , \quad (3)$$

where the operator ^{H} stands for complex conjugate operation and where n is the number of pixels chosen in a boxcar located in each local area.

The averaging operation performed on equation 3, whose main objective is to reduce noise at the expense of losing resolution, mixes values of adjacent pixels, which is not favorable for pixel-wise classification. The averaging algorithm is can be viewed as a non-trainable convolution operation on $\boldsymbol{k}\boldsymbol{k}^H$ with identical kernel weights $\frac{1}{n}$. Letting these kernels to be trainable could enhance the performance of classification and segmentation.

Additionally, the diagonal elements of the coherency matrix are real-valued, which is a desirable property in certain cases, but that has no interest when using CVNNs as they can deal with complex-valued data naturally. Therefore, we propose to use Pauli vector $\boldsymbol{k}$ as CVNN input whenever this data format is available.

To validates our statement, we porpose to evaluate the performance of proposed networks on San Francisco AIRSAR dataset (figure 1a). The ground truth was obtained from [15]. The dataset presents five classes with different occurrences, which are Mountain (7.81%), Water (41.08%), Urban (42.73%), Vegetation (6.67%) and Bare soil (1.71%), as it can be seen on figure 1b.

The sliding window operation [16] was used to generate the dataset with the same parameters used in [7] for stride and window size. With this method, we obtained smaller images patches of size 128×128. We used 80% of the generated patches as training and 10% for each validation and test sets.

3. NETWORK ARCHITECTURES

To this date, difficulties in implementing CVNN models in practice have slowed down the field from growing further [17]. An open-sourced and well documented tool has been developed for this work which enables and facilitates the implementation of a wide range of CVNN architectures [18] for the community to further exploit. This tool also allows to generate from a complex-valued network model its real-equivalent model in terms of trainable parameters while keeping the same architecture with a constant aspect ratio as described in [12].

In this work, we implemented a FCNN architecture using the above mentioned cvnn toolbox [18] inspired by the models in reference [7] for both the real- and complex-valued models, as it is the higher claimed accuracy for this application. The model architecture is shown in figure 2 which is composed of the downsampling or feature extraction part and the upsampling part. The downsampling part presents several blocks (B1, B2, B3, B4, B5 and B6). Each block present two sub-modules which are represented on figure 2 in green and red colours. The upsampling part present blocks B7, B8, B9, B10 and B11 which, in term, are a combination of other two sub-modules, the second one being the same green sub-module present on the downsampling section. The first sub-module (yellow) is a max-unpooling module as explained on [19].

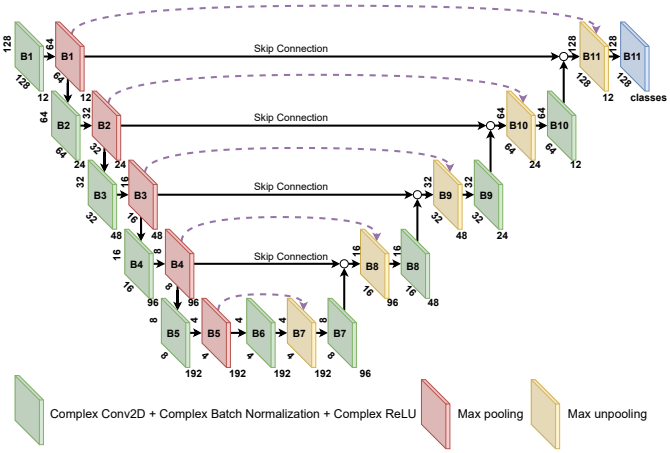


Fig. 2: Chosen Complex-Valued Fully Convolutional Neural Network architecture.

The green sub-module is a combination of a convolution layer, a BatchNormalization (BN) (complex adaptations explained on [20] sections 3.2 and 3.5) and Rectified Linear Unit (ReLU). The convolutional filters present on each layer was of size 3×3 and the number used for each layer is represented on figure 2 for the complex model. For the real case, the amount of kernels were multiplied by $\sqrt{2}$ and rounded to the closest integer number to have a real-equivalent model following the methodology described on [12], under which

it can be shown that both models have an equivalent capacity in terms of real-valued trainable parameters. Thus we can perform a fair comparison between CVNN and RVNN.

The complex-valued activation function $\mathbb{C}\mathrm{ReLU}(.)$ simply consists in applying the well known real-valued function $\mathrm{ReLU}(.)$ to both the real and imaginary part separately so that $\mathbb{C}\mathrm{ReLU}(z) = \mathrm{ReLU}(\mathrm{Re}(z)) + j\,\mathrm{ReLU}(\mathrm{Im}(z))$.

This technique is also known as Type A activation function according to [21]. The red sub-module is a max pooling layer, whose main objective is to shrink the image into smaller ones by keeping only the maximum value within a small window, in our case, of size 2×2. For the complex case, the absolute value of the complex number is used for comparison as proposed in [4]. This layer complements with the max-unpooling sub-module (yellow) which receives the locations where the maximum value was found. The max-unpooling layer enlarges the input image by placing their image pixels according to the locations received from the corresponding max-pooling layer represented by the dashed arrow in figure 2 [19].

The last blocks of the downsampling and upsampling parts (B6 and B11) have some difference with respect to the other blocks. B6 removes the max-pooling layer (red) completely. B11 on the other hand, replaces the ReLU activation function with a softmax activation function to be used for the output layer. As with $\mathbb{C}$ReLU, softmax activation function is applied to both the real and imaginary part separately.

Categorical cross-entropy is used as the loss function which, for the complex network is computed twice, using first the real part and then the imaginary part as the prediction result. An average of the two error values is then calculated to be optimized using Adam [22] optimizer with a learning rate of 0.01. It is worth noticing that pixels without labels (black parts on Figure 1b) are not taken into account for loss computation and neither for the accuracy metric.

4. EXPERIMENT

4.1. Evaluation Methodology

Four simulations were done using both CV-FCNN and RV-FCNN architectures with the two discussed input representations. These 4 simulations were performed 50 times each in order to be able to infer statistical analysis over the results. On each iteration, the train, validation and test set was randomly sampled so no two simulations have the same dataset split. For each training, 400 epochs were made.

The median error was computed as in [23]; if median intervals do not overlap, there is a 95% confidence that their values differ. The confidence interval of the mean is calculated for a confidence level of 99%.

4.2. Simulations Results

Figure 3 shows the validation accuracy evolution over 250 epochs after which the difference of performance can no longer be appreciated on that graph. It can be seen that the complex-valued models converge faster than the real-valued models. However, the final achieved accuracy at epoch 400 can be better appreciated on the box plot in Figure 5. Note that the use of the Pauli vector representation increases the accuracy and achieves a lower variance than using the coherency matrix input for both real and complex architectures.

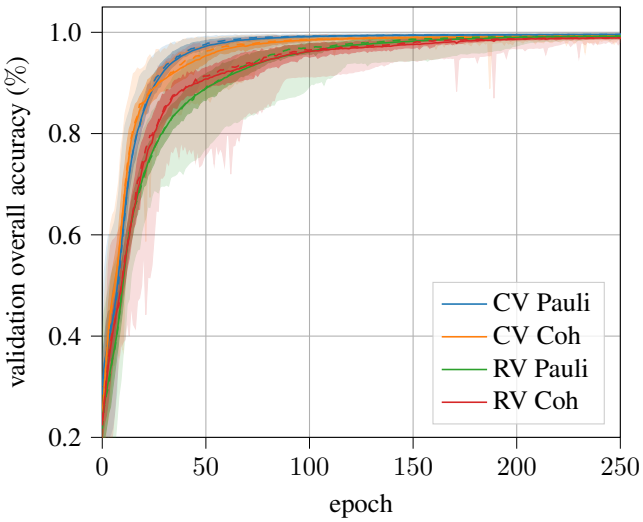


Fig. 3: Validation overall accuracy per epoch.

Statistical indicators of both the Overall Accuracy (OA), which is the ratio of the number of correctly predicted pixels divided by the total number of pixels, and the Average Accuracy (AA), which is an average of the accuracy for each class independently, are summarized in Table 1 which depicts the results obtained with the test set.

The high accuracy obtained mainly for OA makes it harder to discern between the model accuracies, although confidence intervals remains far apart. When comparing the AA, however, the median accuracy between the input representation methods present more than a 1% difference which, above 95%, is highly significant.

Both using a complex-valued architecture and Pauli vector input representation clearly increase accuracy. However, although using a complex-valued architecture may seem to be slightly more significant, the confidence intervals do not allow to assert such conclusions.

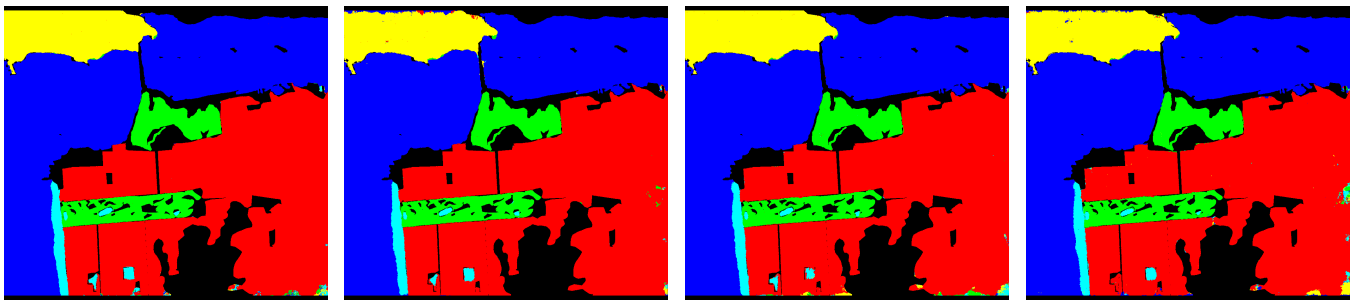
There was a significant difference between the OA and AA. This was due to a highly different classification accuracy per class as it can be appreciated on figure 6. The figure also makes evident that using CVNN achieves a significant amelioration over using RVNN. Finally, figure 4 shows the median predicted image for all models and input representations tested.

5. CONCLUSIONS

In this work, we demonstrated on well-known San Francisco AIRSAR PolSAR database that the CV-FCNN architecture,

		CV-FCNN		RV-FCNN	
		Pauli vector	Coherency matrix	Pauli vector	Coherency matrix
AA	median	**98.00** $\pm$ 0.27	96.80 $\pm$ 0.25	96.75 $\pm$ 0.32	95.20 $\pm$ 0.44
	mean	97.55 $\pm$ 0.15	96.54 $\pm$ 0.12	96.39 $\pm$ 0.18	94.98 $\pm$ 0.21
	full range	93.90 $-$ 98.79	93.44 $-$ 98.63	92.37 $-$ 97.69	91.06 $-$ 97.64
OA	median	**99.64** $\pm$ 0.01	99.45 $\pm$ 0.02	99.40 $\pm$ 0.02	99.19 $\pm$ 0.03
	mean	99.64 $\pm$ 0.01	99.44 $\pm$ 0.01	99.40 $\pm$ 0.01	99.18 $\pm$ 0.02
	full range	99.53 $-$ 99.70	98.91 $-$ 99.61	99.16 $-$ 99.53	98.76 $-$ 99.43

Table 1: FCNN test accuracy results (%).



(a) CV-FCNN Pauli (b) CV-FCNN Coherency (c) RV-FCNN Pauli (d) RV-FCNN Coherency.

Fig. 4: Median predictions for both CV-FCNN and RV-FCNN using both Pauli and coherency matrix representation.

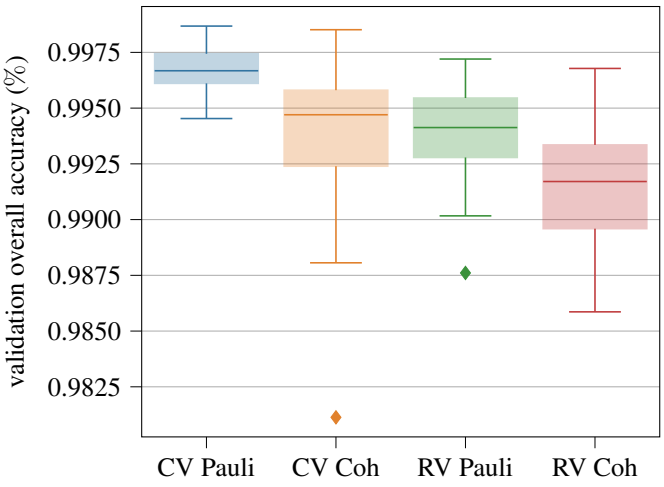


Fig. 5: Validation overall accuracy box plot [24].

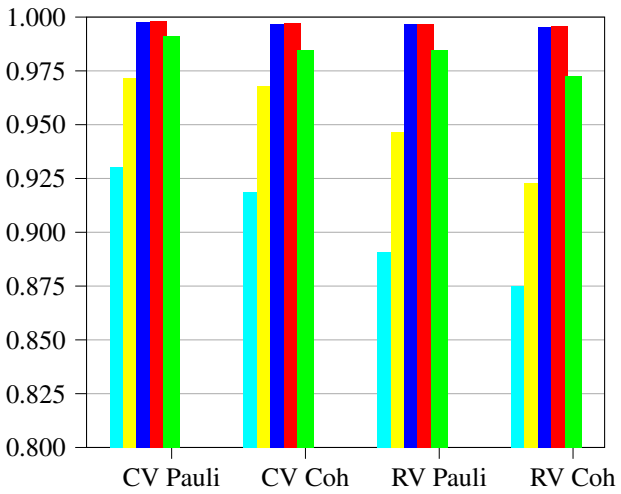


Fig. 6: Test accuracy per class.

implemented using our open-source toolbox [18], achieves a better performance than their equivalent-RV-FCNN for segmentation application.

We also showed that using Pauli vector as input features increases the segmentation performance for both complex-valued and real-valued architectures. We, therefore, encourage the community to favor this form of input representation instead of the widely popular coherency matrix.

Furthermore, it is worth mentioning that the models used in the experiment were dimensioned for a coherency matrix dataset meaning there might be room for improvement for the Pauli vector representation. Yet, it is this last one that performed better. Furthermore, since $\boldsymbol{k}$ is three-dimensional vector. In contrast, the reshape operation of the non-redundant elements of the coherency matrix $\boldsymbol{T}$ leading to a 6th-dimensional vector, the dataset would take less memory space when using the Pauli vector representation.

6. REFERENCES

[1] R. Hänsch and O. Hellwich, "Classification of polarimetric SAR data by complex valued neural networks," in *ISPRS workshop high-resolution earth imaging for geospatial information*, 2009, vol. 38, pp. 4–7.

[2] R. Hänsch and O. Hellwich, "Complex-valued convolutional neural networks for object detection in PolSAR data," in *8th European Conference on Synthetic Aperture Radar*. VDE, 2010, pp. 1–4.

[3] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin, "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1935–1939, 2016.

[4] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.

[5] 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.

[6] Y. Li, Y. Chen, G. Liu, and L. Jiao, "A novel deep fully convolutional network for PolSAR image classification," *Remote Sensing*, vol. 10, no. 12, pp. 1984, 2018.

[7] 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.

[8] H. Parikh, S. Patel, and V. Patel, "Classification of SAR and PolSAR images using deep learning: A review," *International Journal of Image and Data Fusion*, vol. 11, no. 1, pp. 1–32, 2020.

[9] B. Konishi, A. Hirose, and R. Natsuaki, "Complex-valued reservoir computing for interferometric SAR applications with low computational cost and high resolution," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 7981–7993, 2021.

[10] A. Hirose, *Complex-valued neural networks: Advances and applications*, vol. 18, John Wiley & Sons, 2013.

[11] J. Bassey, L. Qian, and X. Li, "A survey of complex-valued neural networks," *arXiv preprint arXiv:2101.12249*, 2021.

[12] J. A. Barrachina, C. Ren, G. Vieillard, C. Morisseau, and J.-P. Ovarlez, "About the equivalence between complex-valued and real-valued fully connected neural networks - application to PolInSAR images," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[14] J. S. Lee and E. Pottier, *Polarimetric radar imaging: from basics to applications*, CRC press, 2017.

[15] X. Liu, L. Jiao, and F. Liu, "PolSF: PolSAR image dataset on San Francisco," *arXiv preprint arXiv:1912.07259*, 2019.

[16] Y. Li, Y. Chen, G. Liu, and L. Jiao, "A novel deep fully convolutional network for PolSAR image classification," *Remote Sensing*, vol. 10, no. 12, 2018.

[17] N. Mönning and S. Manandhar, "Evaluation of complex-valued neural networks on real-valued classification tasks," *arXiv preprint arXiv:1811.12351*, 2018.

[18] J. A. Barrachina, "Complex valued neural networks (cvnn)," Oct. 2021, DOI: 10.5281/zenodo.4452131.

[19] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, *Hands-on convolutional neural networks with TensorFlow: Solve computer vision problems with modeling in TensorFlow and Python*, Packt Publishing Ltd, 2018.

[20] C. Trabelsi, A. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengi, and J. Pal, C., "Deep complex networks," 2018.

[21] 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.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[24] D. F. Williamson, R. A. Parker, and J. S. Kendrick, "The box plot: a simple visual method to interpret data," *Annals of internal medicine*, vol. 110, no. 11, pp. 916–921, 1989.