

Advanced Machine Learning Techniques for Self-Interference Cancellation in Full-Duplex Radios

Andreas Toftegaard Kristensen*, Andreas Burg*, and Alexios Balatsoukas-Stimming[†]

*Telecommunication Circuits Laboratory, École polytechnique fédérale de Lausanne, Switzerland

[†]Department of Electrical Engineering, Eindhoven University of Technology, Netherlands

Abstract—In-band full-duplex systems allow for more efficient use of temporal and spectral resources by transmitting and receiving information at the same time and on the same frequency. However, this creates a strong self-interference signal at the receiver, making the use of self-interference cancellation critical. Recently, neural networks have been used to perform digital self-interference with lower computational complexity compared to a traditional polynomial model. In this paper, we examine the use of advanced neural networks, such as recurrent and complex-valued neural networks, and we perform an in-depth network architecture exploration. Our neural network architecture exploration reveals that complex-valued neural networks can significantly reduce both the number of floating-point operations and parameters compared to a polynomial model, whereas the real-valued networks only reduce the number of floating-point operations. For example, at a digital self-interference cancellation of 44.51 dB, a complex-valued neural network requires 33.7 % fewer floating-point operations and 26.9 % fewer parameters compared to the polynomial model.

I. INTRODUCTION

For beyond-5G communication systems to reach orders-of-magnitude better performance than current systems, a combination of new techniques are required. One such technique is in-band full-duplex (FD), where information is transmitted and received simultaneously and on the same frequency band. While FD systems have long been considered impractical due to the strong self-interference (SI) caused by the transmitter to its own receiver, more recent work on the topic (e.g., [1]–[4]) has demonstrated that it is possible to achieve sufficient SI cancellation to make FD systems viable.

The SI cancellation is usually performed in multiple steps, with the goal of reducing the SI signal to the receiver noise floor. The SI is usually first partially removed in the analog RF domain, applying either passive and/or active suppression to avoid saturating the analog-to-digital converter (ADC) of the receiver. However, analog cancellation is generally expensive due to the additional analog circuitry and a residual SI signal typically still remains at the receiver, which is canceled in the digital domain. This requires modeling the non-linear effects of the different stages of the transceiver, such as digital-to-analog converter (DAC) and ADC non-linearities [5], IQ imbalance [5], [6], phase-noise [7], [8], and power amplifier (PA) non-linearities [4]–[6], [9]. Traditionally, this has been done using polynomial models, which have been shown to work well in practice. However, the polynomial models have

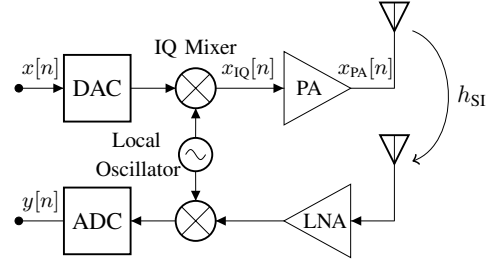


Fig. 1: Simplified wireless FD transceiver block diagram [11].

a high implementation complexity as the number of estimated parameters grows rapidly with the maximum considered non-linearity order. As an alternative to polynomial models, neural networks (NNs) have recently been proposed for SI cancellation [10], [11] where it was shown that NNs can achieve similar SI cancellation performance with a polynomial model with significantly lower computational complexity.

Contribution: In this work, we revisit and extend the work of [10]. More specifically, in addition to real-valued (feed-forward) NNs (RVNNs), we also consider recurrent neural networks (RNNs) and complex-valued neural networks (CVNNs). Moreover, we perform an in-depth network architecture exploration to evaluate the performance of the different NNs as a function of the number of floating-point operations (FLOPs) and the number of NN parameters. This exploration shows that the CVNNs consistently require fewer parameters than their real-valued counterparts for the same SI cancellation performance, indicating that the complex-valued representation is more powerful for the SI cancellation problem. Moreover, we also show that the CVNNs can reduce both the number of FLOPs and parameters compared to a polynomial model, whereas the various real-valued networks (RVNNs) typically only reduce the number of floating-point operations and increase the number of parameters. For example, at an SI cancellation of 44.45 dB, a CVNN requires 33.7 % fewer FLOPs and 26.9 % fewer parameters than the polynomial model, while an RVNN (specifically, a real-valued RNN) requires 30.5 % fewer FLOPs but 69.2 % more parameters.

II. BACKGROUND

In this section, we first describe the main task of digital SI cancellation. Then, we provide an overview of the existing methods for digital SI cancellation and we provide some background on RNNs and CVNNs.

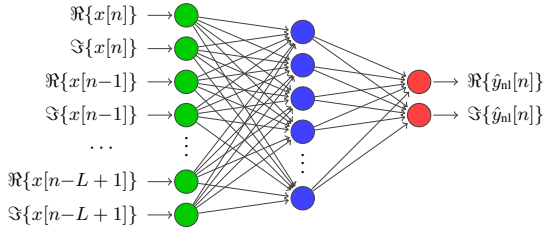


Fig. 2: Example of an FFNN for the reconstruction of the non-linear component of the SI signal [10]. The network is composed of input- (●), hidden- (●), and output-nodes (●).

A. Self-Interference Cancellation

A basic block diagram of an FD transceiver is shown in Fig. 1. If a signal-of-interest from a remote node is not present and assuming that some form of analog SI cancellation is performed, the received signal $y[n]$ is the SI signal that needs to be canceled in the digital domain. The goal of the digital SI cancellation is to reconstruct an estimate of the SI signal $y[n]$, denoted $\hat{y}[n]$, which is a non-linear function of the transmitted baseband samples. The estimated SI signal $\hat{y}[n]$ is then subtracted from $y[n]$ so that the residual SI signal is $e[n] = y[n] - \hat{y}[n]$. Then, the amount of SI cancellation for a window of length N , expressed in dB, is given by $C_{dB} = 10 \log_{10} \left(\frac{\sum_{n=0}^{N-1} |y[n]|^2}{\sum_{n=0}^{N-1} |e[n]|^2} \right)$.

1) *Non-Linear Polynomial SI Cancellation*: For the polynomial SI canceller, the dominant non-idealities to consider are the IQ-mixer and the power amplifier (PA) in the transmitter [4], [9], while the remaining transceiver components are assumed to be ideal. With these assumptions, a state-of-the-art polynomial SI cancellation model is given as [4]

$$y[n] = \sum_{\substack{p=1, \\ p \text{ odd}}}^P \sum_{\substack{q=0 \\ q \text{ odd}}}^p \sum_{l=0}^{L-1} h_{p,q}[l] x[n-l]^q x^*[n-l]^{p-q}, \quad (1)$$

where $x[n]$ is the transmitted digital baseband signal, L corresponds to the overall memory of the system, and P is the non-linearity order. The $\hat{h}_{p,q}$ parameters contain the combined effects from the IQ-mixer, the PA, and the SI channel and they can be obtained using least-squares estimation.

2) *Feed-Forward Neural Network SI Cancellation*: As shown in Fig. 2, a feed-forward NN (FFNN) has three types of nodes: input nodes, hidden nodes, and output nodes. The input to each node of a given layer is a weighted sum of the outputs of the nodes in the previous layer, and the output of each node is obtained by applying a non-linear *activation* function to this weighted sum. The SI signal can be conceptually decomposed as $y[n] = y_{lin}[n] + y_{nl}[n]$, where $y_{lin}[n]$ represents the linear part and $y_{nl}[n]$ the non-linear part. The linear part can be estimated using standard linear cancellation, while the remaining part, y_{nl} , is estimated using the NNs and the two terms are added to re-generate the SI cancellation signal $\hat{y}[n] = \hat{y}_{lin}[n] + \hat{y}_{nl}[n]$. Since the baseband samples are complex-valued, the real and imaginary parts are split and provided as separate inputs to all the RVNNs as shown in Fig. 2.

TABLE I: CVNN activation functions.

Amp-Phase [15]	$f(z) = \tanh(z) \exp(i\theta_z)$
Cardioid [16]	$f(z) = \frac{1}{2}(1 + \cos(\theta_z))z$
modReLU [17]	$f(z) = \max(0, z + b) \exp(i\theta_z)$
CReLU [18]	$f(z) = \text{ReLU}(\Re(z)) + i\text{ReLU}(\Im(z))$
$z\text{ReLU}$ [15]	$f(z) = \begin{cases} z & \theta_z \in [0, \pi/2] \\ 0 & \text{otherwise} \end{cases}$

B. Advanced Neural Networks

1) *Recurrent Neural Networks*: The transceiver chain shown in Fig. 1 have memory, motivating the use of recurrent neural networks (RNNs), a different type of NN architecture more suitable for learning from sequential data. For a fully-connected RNN with a single recurrent layer and one output layer, the computation for each time-step is given by

$$\mathbf{h}[n] = \sigma(\mathbf{W}_h \mathbf{x}[n] + \mathbf{U}_h \mathbf{h}[n-1] + \mathbf{b}_h), \quad (2)$$

$$\mathbf{y}[n] = \mathbf{W}_y \mathbf{h}[n] + \mathbf{b}_y, \quad (3)$$

where $\mathbf{h}[n]$ is the RNN state at time n with input $\mathbf{x}[n]$ and $\mathbf{y}[n]$ is the RNN output. The activation function in the recurrent layer is typically a hyperbolic tangent, $\sigma(x) = \frac{e^{2x}-1}{e^{2x}+1}$. For the RNN, L time-steps of inputs are used to generate the final state $\mathbf{h}[L-1]$, which is then provided to the output layer.

2) *Complex-Valued Neural Networks*: While complex-valued signals can be represented by considering the real and imaginary parts separately, this representation generally does not respect the phase information that is captured by complex algebra. This motivates the use of complex-valued neural networks (CVNNs), where both the network parameters and operations are complex-valued. However, a real-valued scalar loss function of complex variables is not complex-analytic and therefore not complex-differentiable, thus raising the question of how a CVNN can be trained. One solution is to use Wirtinger calculus [12] (or $\mathbb{C}\mathbb{R}$ calculus [13]) to obtain the complex-valued gradients required to train a CVNN.

Let $z \in \mathbb{C}$ and $f(z) \in \mathbb{R}$. The direction of steepest ascent for f is then given as the derivative with respect to z^* , i.e., $\frac{\partial f(z)}{\partial z^*}$ [14]. Using Wirtinger calculus, the derivative $\frac{\partial f(z)}{\partial z^*}$ can be calculated by re-writing $f(z)$ as a bi-variate function of z and z^* , $f(z, z^*)$, and then treat z^* as the variable and z a constant. Alternatively, \mathbb{C} can be regarded as \mathbb{R}^2 , and the complex derivatives can be obtained by considering the partial derivatives with respect to the real and imaginary parts

$$\frac{\partial f}{\partial z} \triangleq \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) \quad \text{and} \quad \frac{\partial f}{\partial z^*} \triangleq \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) \quad (4)$$

Numerous activation functions have been proposed in the literature for CVNNs. Table I shows the different activation functions considered in this work.

III. NETWORK ARCHITECTURE EXPLORATION METHODOLOGY

In this section, we briefly describe the data set and our methodology for the NN architecture exploration.

TABLE II: Mean non-linear cancellation on the test set ± 1 standard deviation for the activation functions using a CVNN with 1 hidden layer of 10 units.

Amp-Phase	Cardioid	modReLU	ℂReLU	zReLU
7.0 \pm 0.3	7.8 \pm 0.1	3.1 \pm 0.3	7.5 \pm 0.2	1.8 \pm 0.4

A. Data Set

The data set consists of QPSK-modulated OFDM signals with a pass-band bandwidth of 10 MHz and $N_c = 1024$ carriers, sampled at 20 MHz. Each transmitted OFDM frame consists of $\sim 20\,000$ baseband samples, with 90% used for training and the remaining 10% used for testing. An average transmit power of 10 dBm is used and the employed two-antenna setup provides a passive analog suppression of 53 dB. Active analog cancellation is not used as the achieved passive suppression is sufficient for this work. The test-bed and the data set are described in more detail in [5] and [10], respectively.

B. Experimental Setup

For the NN experiments, the various NNs are considered for different widths and depths to determine their performance as a function of their computational complexity and memory complexity. We use W to denote the width (i.e., the number of neurons) of a layer. The notation $W-W-W$ is used to indicate an NN with 3 hidden layers and W neurons in each layer. All NNs also have an output layer, with the RVNNs having 2 outputs for the real and imaginary parts and the CVNNs having only 1 complex-valued output. For the FFNNs and RNNs, we consider shallow networks of widths $W = 2, 4, \dots, 20$, and for the deep FFNNs and RNNs we consider the widths $W-W-W = 2-2-2, 4-4-4, \dots, 20-20-20$. For the CVNNs, we consider the sizes $W = 1, 2, \dots, 10$, and for the deep CVNNs we consider the sizes $W-W-W = 1-1-1, 2-2-2, \dots, 10-10-10$. The polynomial model in (1) is considered for powers $P = 3, 5, 7, 9$. For all NNs and the polynomial models, we use $L = 13$ as in the previous work on NNs for SI cancellation with the same data-set [10]. For each of the considered NNs, a hyperparameter search is performed using a grid-search for each NN to select the best values of the learning rate and the batch size. A total of 20 points are sampled from the hyperparameter space, and 5-fold cross-validation is used on the training set for the hyperparameter search with 5 random weight initializations per fold. For the learning-rate, we sample from a continuous uniform distribution $\text{unif}(10^{-6}, 0.05)$ and for the batch-size, we sample from a discrete uniform distribution $\text{unif}(4, 64)$. The NNs are then trained on the entire training set with the best hyperparameters and with 20 different random initializations to see observe the effect of the initialization on the performance. All models are implemented using Tensorflow, which has built-in support for complex-valued operations. All models use the Adam optimizer [19], with the default values for all parameters except for the batch-size and learning-rate, which are selected during the hyperparameter search.

IV. RESULTS

In this section, we present the results of the network architecture exploration. Specifically, we first briefly discuss the results of training CVNNs with different activation functions, we then consider the training behavior of the FFNNs, RNNs, and CVNNs and, finally, we present the SI cancellation performance of the different NNs as a function of the number of FLOPs and parameters.

A. Complex-Valued Activation Functions

Table II shows the results of training the same CVNN with 1 hidden layer of $W = 10$ neurons for the different activation functions given in Table I. The best performance is obtained with the cardioid and ℂReLU activation functions. However, the cardioid function is only marginally better than the ℂReLU function, while being more expensive to compute. The modReLU and zReLU functions have the worst performance, similarly to [18] where the ℂReLU function in all cases performed significantly better than the modReLU and zReLU functions. The Amp-Phase activation function also achieves a relatively high performance level, however, we observed during training that the convergence to a good performance level is much slower than the cardioid and ℂReLU functions. The cardioid and ℂReLU functions reach 6 dB of non-linear cancellation within the first 5 epochs, with the Amp-Phase function requiring 10 epochs. Therefore, in the remainder of this section, we use the ℂReLU function for all CVNNs.

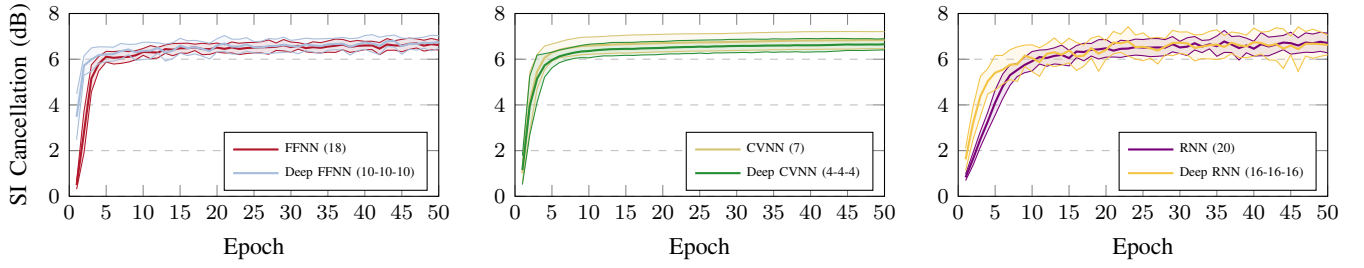
B. Training

In principle, the NNs could learn to provide the full (i.e., linear and non-linear) cancellation, but, as we show in this section, this is difficult in practice. Fig. 4 shows the results of training some shallow and deep FFNNs and CVNNs without first applying a linear SI canceler. The SI linear canceler alone achieves 37.9 dB cancellation and the NNs are unable to achieve a performance much higher than this, even when using deeper networks. Essentially, y_{nl} is significantly weaker than y_{lin} , so the noise in the gradient computation hides the non-linear structure from the learning algorithm.

Fig. 3 shows the performance for the NNs when the linear cancellation y_{lin} is performed separately from the NNs, and the NNs instead learn to cancel only y_{nl} . In this case, the total cancellation achieved is much higher than what is possible when the NNs have to learn to perform the full cancellation. This shows that for the problem of SI cancellation, the inclusion of expert knowledge is essential. We observe that the FFNNs in Fig. 3a and the CVNNs in Fig. 3b achieve their peak performance after a small number of epochs and remain relatively stable, whereas the RNNs in Fig. 3c require more epochs to converge and show a much higher variation, both across different initializations and from one epoch to the next.

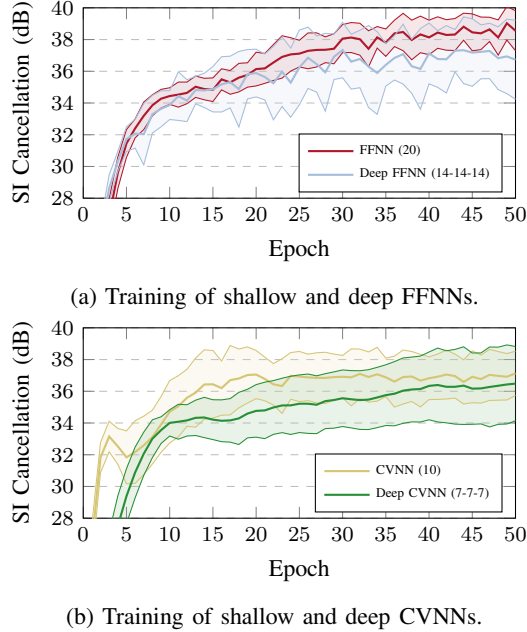
C. Complexity vs Cancellation Performance

In Fig. 5, we present the test set performance of the different NNs trained with the hyperparameters selected from the search. Fig. 5a and Fig. 5b show the total (i.e., linear and



(a) Training of shallow and deep FFNNs. (b) Training of shallow and deep CVNNs. (c) Training of shallow and deep RNNs.

Fig. 3: Mean cancellation on the non-linear test set for 20 different random network initializations ± 1 standard deviation.



(b) Training of shallow and deep CVNNs.

Fig. 4: Mean cancellation on the test set without linear cancellation for 20 different random network initializations ± 1 standard deviation.

non-linear) SI cancellation performance with respect to the number of real-valued FLOPs and the number of real-valued parameters, respectively. The number of real-valued FLOPs and the number of real-valued parameters are a proxy for the computational complexity and the memory requirement of the various SI cancellers, respectively. The number of FLOPs is the sum of the number of equivalent real-valued additions, multiplications, and applications of activation functions. Complex-valued additions and multiplications are converted to real-valued operations assuming that one complex multiplication can be implemented using 3 real multiplications and 5 real additions and one complex addition can be implemented using 2 real additions. For the NN-based cancellers, the number of FLOPs and parameters include the linear canceler. Moreover, to perform a best-case complexity analysis for the polynomial canceler, it is assumed that the calculation of the polynomial basis functions in (1) comes at no computational cost. Finally, we only count the *use* of an activation function, as the specifics

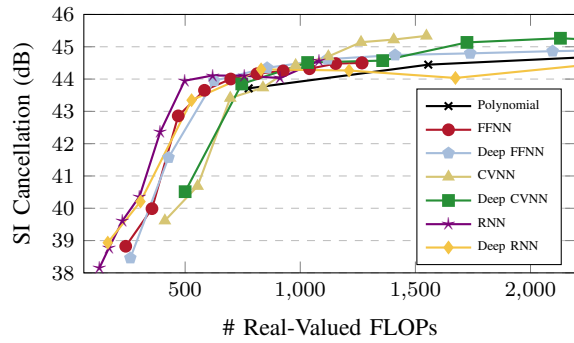
of how exactly an activation function is computed is an implementation detail that is difficult to abstract otherwise.

For the performance as a function of the number of FLOPs, we observe that all the models cluster closely together at an SI cancellation around 44 dB. However, it is clear the CVNNs require much fewer parameters than the other models for the same performance. Finally, the RNNs do not provide significant performance improvements compared to the other models and there is no clear benefit of using deeper NNs.

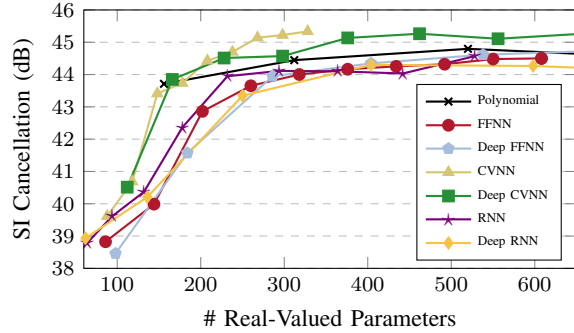
The polynomial model performance peaks around 44.8 dB for $P = 9$, but this requires a very large number of FLOPs that puts it outside the figure. For the sake of comparing a specific performance point, we consider the polynomial model for $P = 5$ that achieves an SI cancellation of 44.45 dB, and then we take the closest NNs which are either equal to or better than the polynomial model in performance. The improvement of the various NN-based cancellers over the polynomial-based canceler is shown in Table III. We observe that the FFNNs reduce the number of FLOPs, but increase the number of parameters significantly. Moreover, the CVNNs are the only NNs capable of reducing both the number of FLOPs and the number of parameters. In general, we can observe from Fig. 5 that for SI cancellation higher than 43.5 dB only the CVNNs remain better than the polynomial model for both the number of FLOPs and the number of parameters, whereas the other NNs generally use more parameters than the polynomial model and only reduce the number of FLOPs.

D. Maximum Cancellation Performance

In Fig. 5, we observe that the CVNNs achieve the highest cancellation performance of all the considered SI cancellers. This indicates that for the problem of SI cancellation, the CVNNs seem to have more representational power. However, the RVNNs still achieve a maximum performance close to that of the CVNNs. As shown earlier, one of the best activation functions for the CVNNs is the $\mathbb{C}\text{ReLU}$ function, which applies the ReLU separately to the real and imaginary parts. It is expected that this type of activation function works well for problems where some symmetry or a special meaning on the real and imaginary parts is required [15]. It is also implied in [15] that a network processing n -dimensional information with activation functions similar to $\mathbb{C}\text{ReLU}$, has neural dynamics similar to that of an RVNN processing



(a) Performance as a function of # real-valued FLOPs.



(b) Performance as a function of # real-valued parameters.

Fig. 5: Test set performance for all models as a function of their computational and memory complexity.

2n-dimensional real-valued information, where the real and imaginary parts are dealt with separately and independently. This may provide a more intuitive explanation as to why the maximum performance of the CVNNs and RVNNs is similar.

V. CONCLUSION

In this paper, we provided an investigation into the trade-offs of applying different NNs at different performance goals for SI cancellation in FD radios. We showed that CVNNs consistently require much fewer parameters than their real-valued equivalents for the same SI cancellation performance, indicating that the complex-valued representation is more powerful for this problem. Moreover, we showed that only CVNNs can reduce both the number of parameters and FLOPs compared to the polynomial SI canceller. The use of the CReLU activation function for the CVNNs also indicates that the neural dynamics of the CVNNs may resemble that of the RVNNs, providing an intuitive explanation as to why the difference in maximum performance is relatively close for the CVNN and RVNNs. Overall, when performing a careful NN architecture exploration combined with hyperparameter tuning, the results are somewhat less favorable (while still being very good in certain cases) than our initial results reported in [10], where we only considered a single NN architecture with fixed hyperparameters and a single SI cancellation performance point. This highlights the importance of a rigorous comparison that is unfortunately often overlooked in the literature that applies NNs to communications problems.

TABLE III: Reduction of # FLOPs and # parameters relative to the polynomial model at an SI cancellation of 44.45 dB. The polynomial model has 1556 FLOPs and 312 parameters.

	Cancellation	# FLOPs	# Parameters
FFNN (18)	44.48 dB	−27.5 %	+76.3%
Deep FFNN (10-10-10)	44.61 dB	−29.8%	+72.4%
CVNN (7)	44.69 dB	−27.9%	−23.7%
Deep CVNN (4-4-4)	44.51 dB	−33.7%	−26.9%
RNN (20)	44.57 dB	−30.5%	+69.2%
Deep RNN (16-16-16)	44.49 dB	+82.4%	+355.1%

REFERENCES

- [1] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, “Practical, real-time, full duplex wireless,” in *ACM MobiCom*, 2011, pp. 301–312.
- [2] M. Duarte, C. Dick, and A. Sabharwal, “Experiment-driven characterization of full-duplex wireless systems,” in *IEEE Trans. Wireless Commun.*, vol. 11, no. 12, Dec. 2012, pp. 4296–4307.
- [3] D. Bharadia, E. McMillin, and S. Katti, “Full duplex radios,” in *ACM SIGCOMM*, 2013, pp. 375–386.
- [4] D. Korpi, L. Anttila, and M. Valkama, “Nonlinear self-interference cancellation in MIMO full-duplex transceivers under crosstalk,” *EURASIP J. on Wireless Comm. and Netw.*, vol. 2017, no. 1, p. 24, Feb. 2017.
- [5] A. Balatsoukas-Stimming, A. C. M. Austin, P. Belanovic, and A. Burg, “Baseband and RF hardware impairments in full-duplex wireless systems: experimental characterisation and suppression,” *EURASIP J. on Wireless Comm. and Netw.*, vol. 2015, no. 142, 2015.
- [6] D. Korpi, L. Anttila, V. Syrjälä, and M. Valkama, “Widely linear digital self-interference cancellation in direct-conversion full-duplex transceiver,” *IEEE J. Selected Areas in Commun.*, vol. 32, no. 9, pp. 1674–1687, 2014.
- [7] A. Sahai, G. Patel, C. Dick, and A. Sabharwal, “On the impact of phase noise on active cancellation in wireless full-duplex,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4494–4510, Nov. 2013.
- [8] V. Syrjala, M. Valkama, L. Anttila, T. Riihonen, and D. Korpi, “Analysis of oscillator phase-noise effects on self-interference cancellation in full-duplex OFDM radio transceivers,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 2977–2990, June 2014.
- [9] L. Anttila, D. Korpi, E. Antonio-Rodríguez, R. Wichman, and M. Valkama, “Modeling and efficient cancellation of nonlinear self-interference in MIMO full-duplex transceivers,” in *IEEE GLOBECOM*. IEEE, 2014, pp. 777–783.
- [10] A. Balatsoukas-Stimming, “Non-linear digital self-interference cancellation for in-band full-duplex radios using neural networks,” in *IEEE Int. Workshop on Signal Proc. Advances in Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.
- [11] Y. Kurzo, A. Burg, and A. Balatsoukas-Stimming, “Design and implementation of a neural network aided self-interference cancellation scheme for full-duplex radios,” in *Asilomar Conf. on Signals, Systems, and Computers*. IEEE, 2018, pp. 589–593.
- [12] W. Wirtinger, “Zur formalen theorie der funktionen von mehr komplexen veränderlichen,” *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, 1927.
- [13] K. Kreutz-Delgado, “The complex gradient operator and the CR-calculus,” *arXiv preprint arXiv:0906.4835*, 2009.
- [14] P. Bouboulis, “Wirtinger’s calculus in general Hilbert spaces,” *arXiv preprint arXiv:1005.5170*, 2010.
- [15] A. Hirose, *Complex-valued neural networks*. Springer Science & Business Media, 2012, vol. 400.
- [16] P. Virtue, X. Y. Stella, and M. Lustig, “Better than real: Complex-valued neural nets for MRI fingerprinting,” in *IEEE Int Conf. on Image Processing (ICIP)*, 2017, pp. 3953–3957.
- [17] M. Arjovsky, A. Shah, and Y. Bengio, “Unitary evolution recurrent neural networks,” in *International Conference on Machine Learning (ICML)*, 2016, pp. 1120–1128.
- [18] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, “Deep complex networks,” *arXiv preprint arXiv:1705.09792*, 2017.
- [19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.