Towards Implementation of Neural Networks for Non-Coherent Detection MIMO systems

Alexis Falempin[1], Julien Schmitt[2], Trung Dung Nguyen[2], Jean-Baptiste Doré[1]
[1] CEA, Leti, Univ. Grenoble Alpes, F-38000 Grenoble, France
[2]VSORA, 13 Rue Jeanne Braconnier, 92360 Meudon, France

***Abstract*—In this paper, we propose the use of quantized neural networks to perform non coherent MIMO detector in sub-TeraHertz (THz) communications. Implementing neural networks is challenging because operations are performed using a high number of bits. This results in slow and energy consuming computations. Then, quantization appears to be essential to consider low latency and energy efficient communication systems. Specifically, in this work, we propose quantizing our designed Neural Network (NN) performing demapping operation. We employ VSORA's digital signal processor (DSP) architecture to perform the quantization. We observe the impact of quantization on the bit error rate. Moreover, we also evaluate the power computation of the proposed DSP regarding our NN. Our simulation results show that we can quantize the weights of the NN to only 6-bits with neglectable degradation on the performance. Besides, we expect achieving high throughput (> 1Gbps), with a peak power consumption of only 0.58W. Thus, the proposed quantization scheme and DSP design allow to achieve high throughput and high energy efficiency.**

***Index Terms*—Sub-THz communications, MIMO, Neural networks, Quantization, DSP, Energy-efficiency**

I. Introduction

Future wireless communications systems such as 6G systems are in the lead of developing ultra low-latency communications, high data rate applications and improved reliability [1]. In addition, a sustainable design and development of transmitters and receivers components are primordial to enable green communications. To achieve such challenges, new cutting-edge technologies are emerging in wireless communications field such as sub-TeraHertz (THz) communications [2], artificial intelligence (AI) [3] and high efficiency digital signal processing (DSP) units. Such systems may exhibit high complexity and hardware implementation issues especially for AI solutions. Then, in this paper, we demonstrate that quantization of neural networks can be used to contribute to the design of low-complexity solutions for sub-THz communications.

Sub-THz communications enable high data rate applications due to the large amount of unused bands [4]. However, sub-THz systems suffer from strong phase impairments due to the inefficiency of oscillators. This issue limits the achievable data rate of theses communication systems by causing notable detection errors. To cope with this major issue, a solution is to propose a design of energy receivers coupled to adequate transmissions schemes [5]. In this paper, we use the same system model and scenario description as in [5].

Besides, the breakthrough of machine learning is now anchored in many fields and it is considerably growing in wireless communications field [3][6]. The use of deep learning solutions allow to solve challenging issues and is then adapted to sub-THz communications. In [5], we developed a neural network demapper (NND) which allows to retrieve transmitted symbols within a MIMO channel from simple data and supervised learning. Nonetheless, future communications systems like 6G are in demand of computation efficiency and cost-effective hardware design. Most of the current AI solutions for wireless communications may exhibit issues such as high complexity, hardware implementation and scalability. For instance, most of implementations relying on TensorFlow [7] or PyTorch [8] do not take model optimizations into consideration. Considering quantization, a NN implemented with the previously cited frameworks uses 32-bits to represents floating point numbers, *i.e.* weights, activation functions are represented using 32-bits. This may lead to performance issues regarding memory size, CPU usage and computation speed.

Thus, in this paper, we investigate the use of quantization to optimize the NND proposed in [5]. Specifically, we employ the VSORA Neural Network (VSNN) simulation platform to enable model processing and weights quantization. This platform allows to run cross-compiled applications on a high level model of the DSP which can be configured with various floating point data, quantization patterns and processing power. Floating-point number is represented by its mantissa and exponent [9]. In VSORA ecosystem, quantization is applied on number of bits used for mantissa and exponent independently. In [10], authors conclude that quantization of NNs is currently limited to 8-bits using TensorFlow and PyTorch. VSNN platform allows to lower the number of bits required to represents the NN weights without degrading the performance. Moreover, we instantiate our NND in a simulated DSP to profile the required number of cycles to perform an inference of the NND. This allows to estimate an achievable throughput.

Numerical simulations show that quantization can be lowered to 6-bits for both mantissa and exponent with a slight performance degradation in terms of Bit Error Rate (BER) which can be neglected. Moreover, we can achieve high throughput (> 1Gbps) with a small amount of arithmetic logic units (ALUs) and multiplication plus accumulation (MAC) units.

The remainder of this paper is organized as follows. Sec. II recalls the communication system used in [5]. We introduce the NND in Sec. III including its architecture and training and inference stages. While Sec. IV presents the solution used for quantization and DSP profiling, Sec. V describes the results of numerical simulations while Sec. VI draws a conclusion with future perspectives.

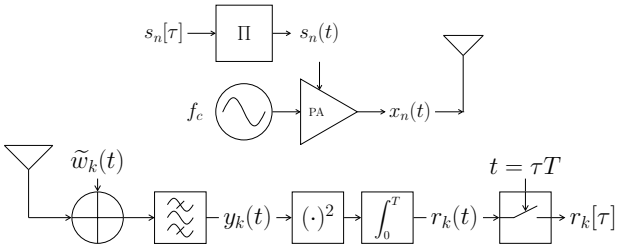

Fig. 1. Block diagram of one Tx-Rx chain

II. System Model

In this section, we consider the system model described in [5]. We consider a non coherent MIMO communication system with N_t transmit antennas and N_r receive antennas with $N_t \leq N_r$. Envelope modulation is considered for the transmitter and the receiver is simply an energy detector.

A. Transmitter RF chain

The transmitter implements envelope modulation and the architecture of one of its RF chains is depicted in Fig. 1. On the n-th RF chain, the signal resulting from a rectangular pulse-shaping Π is defined as follow:

$$s_n(t) = \sum_{\tau \in \mathbb{Z}} s_n[\tau] \cdot \frac{\Pi\left(\frac{t}{T} - \tau - \frac{1}{2}\right)}{\sqrt{T}}, \quad t \in \mathbb{R}, \tag{1}$$

where $s_n[\tau]$ is the τ-th modulated symbol from the On-Off Keying (OOK) constellation $\mathcal{C} = \{0, \sqrt{2}\}$ and T is the symbol duration. We have $\int_{\tau T}^{\tau T + T} |s_n(t)|^2 dt = s_n[\tau]^2$. The transmitted signal $x_n(t)$ at carrier frequency f_c is given by

$$x_n(t) = s_n(t) \cdot \sqrt{2} \cos(2\pi f_c t + \phi(t)), \tag{2}$$

where $\phi(t)$ is a stochastic process modeling a strong oscillator phase noise. The transmitter uses a single oscillator reference, common to all RF chains.

B. Channel model

Sub-THz propagation channels are dominated by a single path, often the LoS direct path, which provides most of the energy contribution [11] [12]. This is due to the usage of directive antennas, sometimes at both transceiver sides. We assume in this work a static LoS channel model.

C. Receiver RF chain

The receiver RF chains architecture is depicted in Fig. 1 for the k-th RF chain. The signal after a band pass filter (with bandwidth $B \geq 2/T$) is given by

$$y_k(t) = \sum_{n=1}^{N_t} h_{k,n} s_n(t) \sqrt{2} \cos\left(2\pi f_c t + \varphi_{k,n} + \phi(t)\right) + w_k(t), \tag{3}$$

where $w_k(t)$ is a band-limited continuous Gaussian process with spectral density N_0, modeling the thermal noise and $h_{k,n}$ the channel coefficient between TX antenna k and Rx antenna n.

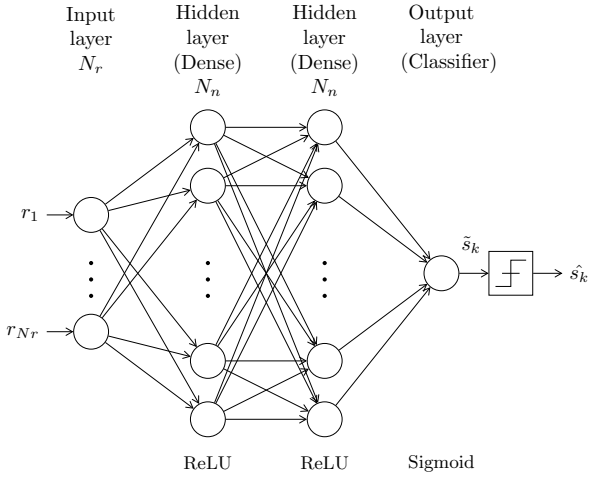

Fig. 2. Architecture of the k-th neural network of the NND

The received symbols are given by [5],

$$r_k[\tau] = E_k[\tau] + \sqrt{2E_k[\tau]} \cdot w_k[\tau] + z_k[\tau], \tag{4}$$

where $w_k[\tau] \sim \mathcal{N}(0, \sigma_w^2)$ is a zero-mean Gaussian variable and $z_k[\tau]$ is a chi-square distributed variable. In case of ideal directive antennas, the co antenna interference is null and the message can be restored. In this work, we assume that interference exists. The complete analysis of such a system has been conducted in [5] and the benefits of considering neural network based detectors have been highlighted.

III. Neural Network Demapper (NND)

As proposed in [5], using a NN as a demapper allows to propose new perspectives for detection. Indeed, such a solution does not explicitly need the channel propagation matrices to perform symbol estimation. Moreover, it also allows to run symbol detection without any assumption on the channel which is mandatory for Maximum Likelihood Detection under Gaussian Approximation (MLD-GA) algorithm used in previous work.

A. Architecture

The NND is composed of multiple neural network to estimate transmitted symbols. Each of the N_t NN estimates one transmitted symbol. The architecture of the k-th NN is presented on Fig. 2. The neural network is composed of N_{hl} fully connected layers. Each layer has N_n neurons and uses Rectified Linear Unit (ReLU) as activation function. The input layer contains N_r entries each one representing a received symbol r_k. Finally, the NN outputs the prediction $\widetilde{s}_k$, homologous to the probability $P_r(s_k = 0|\mathbf{r})$ using a single neuron with sigmoid activation. Besides, each neural network can be trained separately, transmitted symbols can be estimated independently.

B. Training

Each neural network of the NND is trained independently using reference symbols known at the receiver. Since OOK signalling is used, the NN learns how to retrieve the transmitted symbol using a binary representation of the symbol. Thus,

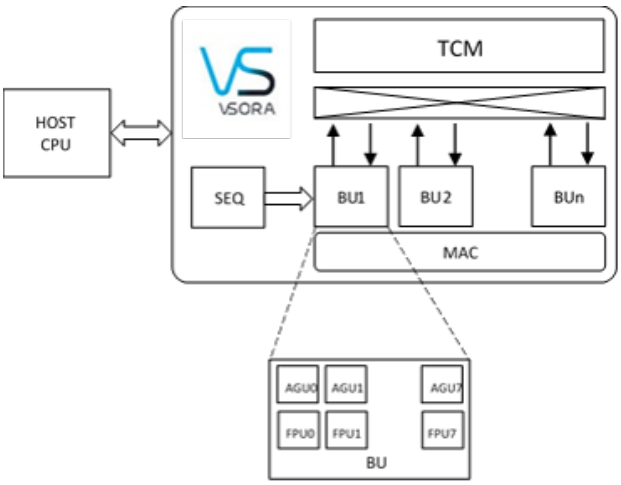

Fig. 3. Architecture of VSORA DSP

we choose to optimize a Binary Cross-Entropy loss J_k using an Adam optimizer. This loss J_k is described by,

$$-\frac{1}{B_s}\sum_{\tau=1}^{B_s}\left[\frac{s_k[\tau]}{\sqrt{2}}\ln\left(\widetilde{s}_k[\tau]\right)+\left(1-\frac{s_k[\tau]}{\sqrt{2}}\right)\ln\left(1-\widetilde{s}_k[\tau]\right)\right], \tag{5}$$

where B_s denotes the batch size used during the training.

C. Inference

During inference or online usage, each neural network of the NND is considered as a static function applied to the received symbols to retrieve each transmitted symbol s_k. In the scenario presented in Sec. V-A, the wireless link is fixed and the channel may vary very slowly or be even static which does not motivate an online adaptation.

IV. DSP ARCHITECTURE

A. Architecture overview

VSORA's DSP architecture is a single instruction multiple data (SIMD) processor. It is composed of many floating point units (FPUs), all performing the same operation $c = f(a, b)$, where a and b are operands of function f and c is the result, sent by a main sequencer (SEQ) as depicted in Fig. 3. A FPU executes floating point operations. It is associated to an address generator unit (AGU) in charge of generating addresses of operands a, b (read process) and c (write process) which are stored in a tightly coupled memory (TCM).

8 FPUs are gathered in a component called base unit (BU). VSORA's DSP IP is scalable: the number N_{BU} of BUs can be chosen between 1 and 64 (8 – 512 FPUs), targeting a wide range of applications with different capabilities of processing power. Tensor mapping in TCM is optimized in order to feed the FPUs with data each cycle, reaching a rate of use up to 90% for computations such as tensor multiplication. In addition to the FPUs, a block containing MAC operators (Multiplication / Accumulation) has access to all data allowing matrix block multiplication. The number of MAC is configurable using single bloc or multi-blocs and can reach up to $64N_{BU}^2$ MACs per core.

B. Development flow

VSORA neural network development flow (VSNN) is organized around simulation platforms:

- The native platform does not have any representation of the hardware and is used to develop the algorithms and the neural network. It is based on standard frameworks (like TensorFlow, ONNX, Caffe ...) associated with a mathematical library developed by VSORA (tensor computation).
- The TLM platform (Transaction Level Model) is a high level representation of the DSP: the model of the DSP is executed on the PC with the application (same code than the native platform) compiled for the target (cross compilation). This platform gives fine estimations of the CPU load and of the memory usage. It is also used for study since it is easy to test a variety of quantization patterns and processing power (number of BUs).
- The register level model (RTL) platform is a low level representation of the DSP and is used for the synthesis of the silicon. It uses the same embedded code as the TLM platform.

C. Data representation

Data are floating point values respecting the IEEE 754 principles, including the denormalization but without specific numbers (infinity, NaN, *etc*). A word is composed of a mantissa of m bits, an exponent of e bits and a sign bit which defines a quantization $q(e, m)$.

Conventional quantization schemes used in CPU are float (32-bits) (quantization $q(8, 23)$) or double (64-bits) (quantization $q(11, 52)$). VSORA's DSP IP can be configured with a dedicated quantization to optimize the silicon area and the power consumption. With the TLM platform, it is easy to study the impact of various quantizations such as $q(3, 3)$ (7-bits), $q(4, 3)$ (8-bits), $q(3, 4)$(8-bits), *etc* without being stuck to a fixed quantization like TensorFlow Lite or PyTorch which is limited to 8 bits. For example, the $q(3, 3)$ pattern characteristics are (i) Lowest denormalized number : 2^{-5}, (ii) Dynamic range : $[-15, 15]$.

D. Compilation process

The compilation process is based on a custom compiler (LLVM based). The system code is implemented in high-level language, using, for example, C++ or Matlab-like / Tensorflow-like code. The code is written as if everything is executed on the host. During compilation, the smart compiler will separate the code running on the VSORA DSP from the code running on the host. For the designer it will look as if everything is executed on the host processor. Using one code for both algorithms and the embedded software makes the design process faster and simpler, which results in a minimal learning curve. For neural network codes, a compiler add-on processes the different layers and extracts the corresponding weights.

E. Quantization of Neural Network

VSORA toolchain supports three modes of quantization: Post Training Quantization (PTQ), Light Quantization Aware

TABLE I
SIMULATION PARAMETERS

Parameters	Notation	Values
Carrier frequency	f_c	145 GHz
Symbol rate	$1/T$	1 GHz
Bandwidth	$B = 2/T$	2 GHz
Thermal noise	N_0	-174 dBm/Hz
Noise figure	N_f	10 dB
Antenna gain	g_0	32 dBi
Beam width	θ	3 °
Side lobe level	ϵ	-20 dB
Distance Tx - Rx	d_0	10 m

Training (LQAT) and Quantization Aware Training (QAT). Using PTQ, we directly apply quantization on the pre-trained weights. To reduce accuracy loss, one may need to fine tune the network using a small part of dataset (LQAT mode) or retrain completely with back propagation (QAT mode). PTQ is the fastest but less accurate whereas QAT is the heaviest but most accurate scheme. In this paper, we investigate on PTQ mode.

V. NUMERICAL SIMULATIONS

In this section, we present numerical simulations to evaluate the performance of the proposed quantization for our NN demapper. First, we present the simulation scenario, inspired from [5] which will be used therafter. Next, we analyse the quantization effect on the NND performance. Eventually, we give an estimation of the achievable throughput using our quantized NN in a simulated DSP unit by the VSORA software.

A. Scenario description

We consider a point to point wireless link in the D-band. Table I describes the simulation parameters. A uniform linear array (ULA) of antennas with $N_t = N_r = N = 4$ is assumed and their specifications are extracted from [13].

Based on the definition of [5], we denote by κ the maximum number of transmitted symbols strongly interfering on a receive antenna. By means of illustration, the channel gain matrix $|\boldsymbol{H}|$ for $N = 4$ and $\kappa = 5$ may be accurately approximated as follows

$$|\boldsymbol{H}| \simeq |h_{1,1}| \cdot \begin{bmatrix} 1 & 1 & 1 & \rho \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \rho & 1 & 1 & 1 \end{bmatrix}, \tag{6}$$

where ρ is the residual interference due to side lobes of the antennas with $\rho^2 = -40$ dB. There are κ diagonals, whose elements are 1, which equals the maximum number of interfering symbols on a receive antenna.

B. Performance assessment

We evaluate the above scenario for a specific case to analyze the proposed quantization and throughput estimation. Thus, here we choose a MIMO system with $N = 4$ and $\kappa = 5$ resulting in strong spatial interference.

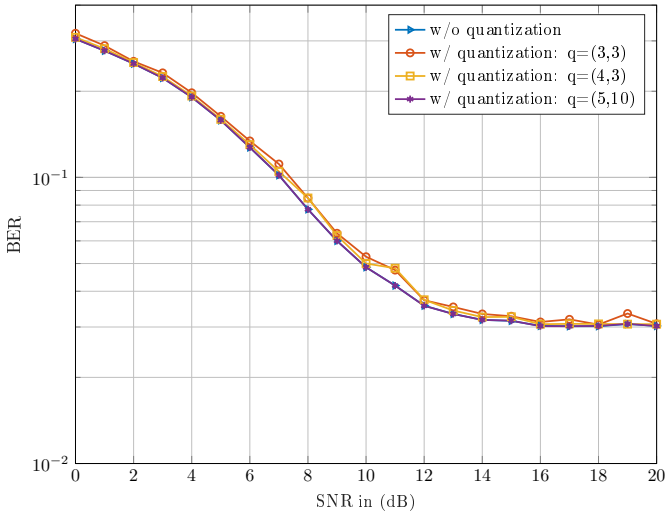

Fig. 4. Impact of quantization on the BER

1) Post training quantization: Using VSORA solution and workflow, we perform a post training quantization on the NND, *i.e.* we apply a quantization on the weights learned during the training phase. Several quantization modes are tested. We denote by q the quantization mode used. As stated before in Sec. IV, a floating-point number can be represented using mantissa and exponent bits.

Fig. 4 presents the impact of quantizing NND weights on the Bit Error Rate (BER) performance. First, it must be noted that the BER performance without quantization, refered to as the boundary, is performed using TensorFlow with 32bits floating-point number representation. Next, we can observe that using $q = (5, 10)$, we obtain the same performance as boundary. Using $q = (4, 3)$ and $q = (3, 3)$ will induce a slight degradation which can be neglected.

2) Throughput v.s. Processing power: In addition to quantization considerations, we perform a throughput estimation using VSORA's DSP. To estimate the achievable throughput, we count the number of cycles required to perform a batched inference of received symbols in function of the DSP resources. The DSP resources are quantified by the number of BUs, *i.e.* the number of MAC operators. In our case, the optimal number of MAC operators, $N_{MAC} = 128N_{BU}$. Besides, one can derive the processing power

$$P_p = 2N_{MAC}F_{clk}, \tag{7}$$

where F_{clk} is the clock frequency of the processor. One MAC is equivalent to 2 Floating Point Operations. Thus, P_p can be expressed in Floating Point Operations per second (FLOPS). Usually, it is expressed in TeraFLOPS (TFLOPS) since high-frequency processors are employed.

In our simulation, we consider a clock frequency $F_{clk} =$ 2GHz of the processor. This allow us to derive the achievable throughput. Fig. 5 presents the theoretical throughput we can achieve regarding the processing power. It must be underlined that the provided estimation has been done using a single core for brevity.

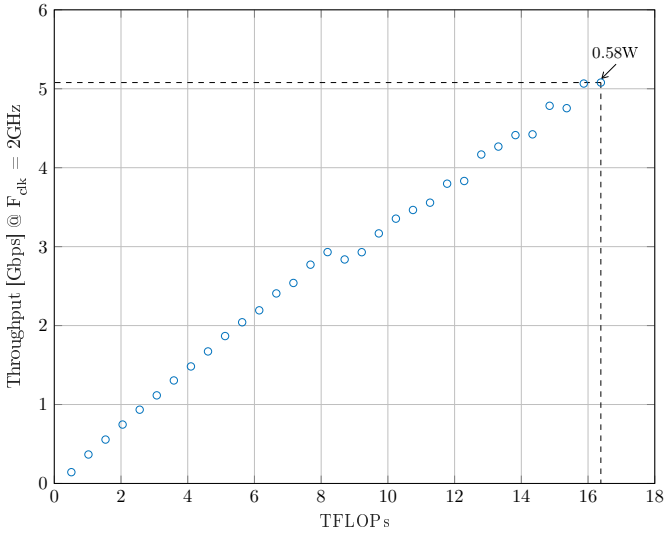

Fig. 5. Achievable throughput in Gbps in function of the processing power

Besides, an estimation of the power consumption can be estimated from the simulation environment. If we consider $N_{MAC} = 4096$, leading to $\sim$ 16TFLOPS, the peak power consumption, *i.e.* when all the hardware resources are used, is approximately 0.58W. This is equivalent to 0.11nJ/bit for a corresponding throughput of 5Gbps.

C. Discussion

In this paragraph, we assess VSORA's DSP against some state-of-the-art solutions in terms of quantization peak power v.s. processing power. First, in terms of quantization, we use a floating-point quantization up to 6-bits whereas other solutions mainly uses int8 quantization [7][14].

Moreover, VSORA's DSP is proposed as an IP and it is possible to choose several parameters: number of computing elements, number of MACs, quantization, TCM memory size. For a dedicated application we can reach an optimum according to the constraints we have to fulfill and since the problem is multidimensional, this optimum could not be trivial. Regarding our proposed application, the main constraint would be maximizing the throughput while optimizing the power consumption. In [14], authors have made a comparison of several solutions in terms of processing power v.s. peak power. The proposed DSP, compared to solutions in the cited work, exhibits the best trade off between processing power and peak power.

Besides, VSORA's development flow offers a complete toolkit to make the best decision and choose the right architecture.

VI. Conclusion

In this paper, we studied the use of quantization for a neural network performing demapping for sub-THz communications. Quantization is primordial in such systems since low-latency, high throughput and sustainable systems are required. To perform quantization, we used the VSNN platform to apply PTQ to the NND parameters. Simulation results showed that we can lower the quantization of the weights from 32-bits to only 7-bits with no significant performance loss. Using VSNN, we can achieve a lower quantization than native TensorFlow quantization scheme, limited to 8-bits.

Moreover, using VSORA DSP, we achieve high troughput ($>$ 5Gbps) for a peak power of only 0.58W leading to 0.116nJ/bit. This demonstrates that our proposed quantization scheme and DSP achieves high data rate with low-energy consumption. Thus, our proposal respects a high performance and low-energy design required in future wireless communications sytems.

Finally, future work will involve reducing the quantization while enabling QAT to reduce the impact of the quantization on the performance.

Acknowledgement

This work has been funded by the H2020-ECSEL CPS4EU European project (grant agreement 826276).

References

[1] E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret *et al.*, "6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 42–50, Sep. 2019.

[2] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal *et al.*, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.

[3] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari *et al.*, "Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 214–222, 2020.

[4] J.-B. Doré, Y. Corre, S. Bicais, J. Palicot, E. Faussurier, D. Ktenas *et al.*, "Above-90GHz Spectrum and Single-Carrier Waveform as Enablers for Efficient Tbit/s Wireless Communications," in *25th Int. Conf. on Telecommunications (ICT)*, 2018, pp. 274–278.

[5] S. Bicaïs, A. Falempin, J.-B. Doré, and V. Savin, "Design and Analysis of MIMO Systems using Energy Detectors for Sub-THz Applications," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.

[6] A. Salh, L. Audah, N. S. M. Shah, A. Alhammadi, Q. Abdullah, Y. H. Kim *et al.*, "A Survey on Deep Learning for Ultra-Reliable and Low-Latency Communications Challenges on 6G Wireless Systems," *IEEE Access*, vol. 9, pp. 55 098–55 131, 2021.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[8] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," pp. 8024–8035, 2019.

[9] "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019.

[10] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," 2021.

[11] L. Pometcu and R. D'Errico, "Characterization of Sub-THz and mmWave Propagation Channel for Indoor Scenarios," in *12th European Association on Antennas and Propagation (EurAAP)*, Apr 2018.

[12] T. Xing and T. S. Rappaport, "Propagation Measurement System and Approach at 140 GHz–Moving to 6G and Above 100 GHz," in *IEEE Global Communications Conf. (GLOBECOM)*, Dec 2018.

[13] F. F. Manzillo, A. Clemente, and J. L. Gonzalez-Jiménez, "High-gain D-band Transmitarrays in Standard PCB Technology for Beyond-5G Communications," *IEEE Trans. Antennas Propag.*, pp. 1–1, 2019.

[14] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "AI Accelerator Survey and Trends," in *2021 IEEE High Performance Extreme Computing Conf. (HPEC)*, 2021, pp. 1–9.