

# EFFICIENT QUANTIZATION-AWARE NEURAL RECEIVERS: BEYOND POST-TRAINING QUANTIZATION

SaiKrishna Saketh Yellapragada<sup>\*</sup>, Esa Ollila<sup>\*</sup>, Mário Costa<sup>†</sup>

<sup>\*</sup>Aalto University, Espoo, Finland

<sup>†</sup>Nokia Technologies, Amadora, Portugal

{saikrishna.yellapragada, esa.ollila}@aalto.fi, mario.costa@nokia.com

## ABSTRACT

As wireless communication systems advance toward Sixth Generation (6G) Radio Access Networks (RAN), Deep Learning (DL)-based neural receivers are emerging as transformative solutions for Physical Layer (PHY) processing, delivering superior Block Error Rate (BLER) performance compared to traditional model-based approaches. Practical deployment on resource-constrained hardware, however, requires efficient quantization to reduce latency, energy, and memory without sacrificing reliability. In this paper, we extend Post-Training Quantization (PTQ) by focusing on Quantization-Aware Training (QAT), which incorporates low-precision simulation during training for robustness at ultra-low bitwidths. In particular, we develop a QAT methodology for a neural receiver architecture and benchmark it against a PTQ approach across diverse 3GPP Clustered Delay Line (CDL) channel profiles under both Line-of-Sight (LoS) and Non-LoS (NLoS) conditions, with user velocities up to 40 m/s. Results show that 4-bit and 8-bit QAT models achieve BLERs comparable to FP32 models at a 10% target BLER. Moreover, QAT models succeed in NLoS scenarios where PTQ models fail to reach the 10% BLER target, while also yielding an 8 $\times$  compression. These results with respect to full-precision demonstrate that QAT is a key enabler of low-complexity and latency-constrained inference at the PHY layer, facilitating real-time processing in 6G edge devices.

**Index Terms**— Deep Learning, Neural Receivers, quantization aware training, post training quantization, resource efficient machine learning

## 1. INTRODUCTION

Deep Learning has recently delivered strong gains at the PHY, where fully-convolutional neural receivers such as DeepRx replace channel estimation, equalization, and demapping to improve error-rate performance under standardized channels

and mobility [1]. End-to-end designs further demonstrate that learned receivers can reduce or even eliminate pilot overhead while maintaining reliability, highlighting the practical promise of Artificial Intelligence (AI)-native PHY stacks for beyond-5G and 6G systems [2]. However, deployment in edge-constrained radio hardware remains limited by compute, memory, and latency budgets, making precision reduction via quantization central to practical neural receiver inference. In this context, QAT adapts the model during training to low-precision arithmetic [3, 4, 5, 6, 7, 8, 9].

PTQ avoids retraining and compresses models by up to 8 $\times$ , but can degrade at aggressive bit-widths and under distribution shifts characteristic of high-Doppler channels. In prior work, a CNN-based neural receiver with symmetric uniform PTQ (per-tensor and per-channel) achieved near-`float32` BLER at 8-bit and competitive results at 4-bit, establishing a strong baseline for efficient PHY inference and motivating robustness analysis at low precision [10, 11, 12, 13]. In this paper, we examine whether QAT closes the remaining gap by training with simulated quantizers, thereby improving the dependability of ultra-low-bit neural receivers across mobility and channel conditions.

QAT integrates simulated quantizers into the training graph so the network learns to compensate for rounding and range effects, and is widely shown to preserve accuracy at 8-bit and enable viable 4-bit operation across diverse architectures and hardware back ends. Concretely, QAT co-designs scale and rounding behavior with task loss, retaining integer-arithmetic-only inference paths at deployment time to meet edge latency and energy targets without sacrificing model quality. These properties make QAT a natural fit for neural receivers operating under time and frequency-selective fading, where amplitude spikes and distribution shifts can otherwise erode PTQ accuracy at ultra-low bit-widths.

This paper advances efficient neural receivers beyond PTQ by developing a QAT pipeline for convolutional PHY receivers. The principles discussed in this paper should apply to any convolutional-ResNet architectures. We evaluate on 3GPP CDL-B/D channels across low, medium, and high UE speeds using Sionna, a hardware-accelerated differentiable

The work of first author has been supported in parts by Research Council of Finland (grant no:359848 and 6GARROW project: No. RS-2024-00435652).

link-level setup [14].

## Main contributions

- A principled QAT formulation with learnable clipping and per-channel scales, demonstrating that 4-bit QAT preserves deployment efficiency while maintaining accuracy.
- A comprehensive link-level evaluation contrasting PTQ and QAT on 3GPP TR 38.901 CDL-B/D channels across User Equipment (UE) speeds, with results reported at BLER targets of 10% and 1%.

## 2. QUANTIZATION-AWARE TRAINING

QAT simulates low-precision inference during training by inserting fake-quantization operators in the forward pass, allowing the model to adapt to quantization noise and maintain higher accuracy after deployment. This allows for gradient-based optimization of both the model weights and the quantization parameters, despite the discrete nature of the quantization function. The process can be broken down into three distinct steps: clipping, quantization, and dequantization.

Let  $b$  denote the bit-width and  $[\alpha, \beta]$  be the learnable clipping interval. For signed symmetric quantization, the integer range is  $[q_{\min}, q_{\max}]$ , where  $q_{\min} = -2^{b-1}$  and  $q_{\max} = 2^{b-1} - 1$ .

**1. Clipping:** First, an input real-valued  $x$  is clipped to the learnable dynamic range:

$$x_c = \text{clamp}(x, \alpha, \beta) = \max(\alpha, \min(x, \beta)). \quad (1)$$

**2. Quantization:** The clipped value is then quantized to an integer  $q$  using a uniform step size, or scale,  $s$ :

$$s = \frac{\beta - \alpha}{q_{\max} - q_{\min}}, \quad \text{and} \quad q = \left\lfloor \frac{x_c}{s} \right\rfloor. \quad (2)$$

For symmetric quantization, the zero-point is implicitly zero.

**3. Dequantization:** Finally, the integer  $q$  is mapped back to the real domain to produce the output of the fake-quantization operator,  $F_b(\cdot)$ :

$$F_b(x; \alpha, \beta) = s \cdot q. \quad (3)$$

In the forward pass of QAT, all activations and weights are replaced by the output of this operator. For the backward pass, the non-differentiable rounding function  $\lfloor \cdot \rfloor$  is handled using the Straight-Through Estimator (STE) [15, 16]. This allows gradients to be propagated through the operator using the following approximations for the partial derivatives:

$$\frac{\partial F_b}{\partial x} \approx \mathbb{1}_{\{\alpha \leq x \leq \beta\}}, \quad \frac{\partial F_b}{\partial \alpha} \approx \mathbb{1}_{\{x < \alpha\}}, \quad \frac{\partial F_b}{\partial \beta} \approx \mathbb{1}_{\{x > \beta\}}. \quad (4)$$

The gradient with respect to  $x$  passes through unchanged only for values within the clipping range. The gradients for  $\alpha$  and

$\beta$  are non-zero only for clipped values, which serves to penalize the clipping bounds and encourages them to expand to fit the dynamic range of the inputs. This allows for the end-to-end training of both network and quantization parameters.

### 2.1. Training Objective Under Simulated Quantization

The training objective in QAT is to minimize the task loss when model weights operate under quantized precision while maintaining full-precision activations. The neural receiver function  $f(\mathbf{s}; W_q)$  denotes the forward computation that processes an input signal  $\mathbf{s}$  using quantized weights  $W_q = F_b(W; \phi_w)$  to produce the network prediction. Using the fake quantization operator  $F_b(\cdot)$  defined in (3), the optimization problem becomes:

$$\min_{W, \phi_w} \mathbb{E}_{(\mathbf{y}, \mathbf{s}) \sim \mathcal{D}} [\mathcal{L}(\mathbf{y}, f(\mathbf{s}; F_b(W; \phi_w)))] \quad (5)$$

where  $W$  denotes the network weights,  $\phi_w$  represents the quantization parameters for weights (clipping bounds  $\alpha$  and  $\beta$ ), and  $\mathcal{D}$  is the training data distribution with signal samples  $\mathbf{s}$  and labels  $\mathbf{y}$ . Note that in this work, activations remain in full precision throughout the network, allowing the neural receiver to preserve signal fidelity while achieving computational efficiency through weight quantization.

During quantization-aware fine tuning, weight updates are applied to full-precision master copies of  $W$ , ensuring proper gradient accumulation despite quantization noise. The quantization parameters  $\phi_w$  are jointly optimized with the network weights to learn optimal dynamic ranges that minimize the quantization-aware loss.

## 3. QUANTIZATION-AWARE NEURAL RECEIVER

### 3.1. System Model

We consider an uplink Single-Input-Multiple-Output (SIMO) Orthogonal Frequency Division Multiplexing (OFDM) system with  $N_{\text{Rx}}$  receive antennas. The input bitstream is Low-Density Parity-Check (LDPC)-encoded, mapped to modulation symbols, and arranged into a Resource Grid (RG) of size  $N_{\text{sym}} \times N_{\text{sc}}$ , with Demodulation Reference Signals (DMRSs) embedded at known time-frequency locations for channel estimation. After Inverse Fast Fourier Transform (IFFT) and cyclic prefix insertion, the signal is transmitted over the 3GPP CDL channel [17]. At the receiver, Fast Fourier Transform (FFT) yields

$$\mathbf{y}_{n,k} = \mathbf{h}_{n,k} x_{n,k} + \mathbf{n}_{n,k}, \quad \mathbf{y}_{n,k}, \mathbf{h}_{n,k} \in \mathbb{C}^{N_{\text{Rx}} \times 1}, \quad (6)$$

where  $x_{n,k}$  is the transmitted symbol,  $\mathbf{n}_{n,k} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_{\text{Rx}}})$ , and  $\mathbb{E}[|x_{n,k}|^2] = 1$ . Channel estimates  $\mathbf{h}_{n,k}$  at pilot positions are interpolated across the RG. Traditionally, when the receiver is processing the OFDM waveform, the post-FFT waveform at the receiver is fed to perform channel estimation, equalization, and demapping.

**Table 1: Training Parameters and Randomization**

Parameter	Training	Randomization
Carrier Frequency	3.5 GHz	None
Channel Model	CDL-[A,C,E]	Uniform
RMS Delay Spread	10–100 ns	Uniform
UE Velocity	0–50 m/s	Uniform
SNR	−2–15 dB	Uniform
Subcarrier Spacing	30 kHz	None
Modulation Scheme	64-QAM	None
No. of Tx Antennas	1	None
No. of Rx Antennas	2	None
Code Rate	0.5	None
DMRS Configuration	3 <sup>rd</sup> , 12 <sup>th</sup> symbol	None
Optimizer	Adam	None
Batch Size	128	None

**Table 2: Testing Parameters**

Parameter	Tested on
Channel Model	CDL-[B,D] Low: 0–5.1 m/s
UE Velocity	Medium: 10–20 m/s High: 25–40 m/s
SNR	0–12 dB

### 3.2. Neural receiver

We consider a neural receiver architecture and the training strategy based on our previous work [10]. This is designed to replace traditional signal processing operations where the input is the post-FFT sequence and the output is Log-Likelihood Ratios (LLRs). The output of the network which are the LLRs, are then used as input to LDPC decoding. The neural receiver is trained with the parameters mentioned in Table 1 using channel models CDL-A, C, and E. This training scenario is a combination of NLoS and LoS models, while testing is performed using CDL-B and CDL-D.

The neural receiver is trained to maximize the bit-metric decoding performance [18], using Binary Cross-Entropy (BCE) loss for quantifying the difference between the network’s predicted LLRs and the true coded bits across the complete OFDM resource grid. The BCE loss-function is

$$\mathcal{L}_{\text{BCE}}(B, \hat{L}) = -\mathbb{E} \left[ B \log \sigma(\hat{L}) + (1 - B) \log(1 - \sigma(\hat{L})) \right], \quad (7)$$

where  $B$  denotes the ground-truth coded transmitted bits,  $\hat{L}$  denotes the predicted LLRs and  $\mathbb{E}[\cdot]$  expectation operator. The function  $\sigma(\cdot)$  represents the sigmoid activation, which maps LLRs to probabilities.

### 3.3. QAT Fine-Tuning Protocol

Building upon the baseline neural receiver trained with full-precision weights, we implement QAT fine-tuning to recover performance degradation observed in our previous PTQ analysis. The QAT process begins by initializing the fake quan-

tization operators  $F_b(\cdot)$  defined in (3) throughout the neural receiver architecture. The quantization parameters  $(\alpha, \beta)$  for each quantized layer are initialized based on the weight distributions of the full-precision baseline model to ensure stable convergence during fine-tuning.

The quantization-aware fine-tuning employs a conservative training regime designed to adapt the neural receiver to quantization noise while preserving the learned signal processing capabilities. After inserting simulated quantization operations into the neural receiver architecture, the model undergoes fine-tuning for up to 5000 epochs with a reduced learning rate of  $10^{-6}$ . This significantly lower learning rate compared to initial training prevents catastrophic disruption of the pre-trained representations while allowing gradual adaptation to quantization constraints.

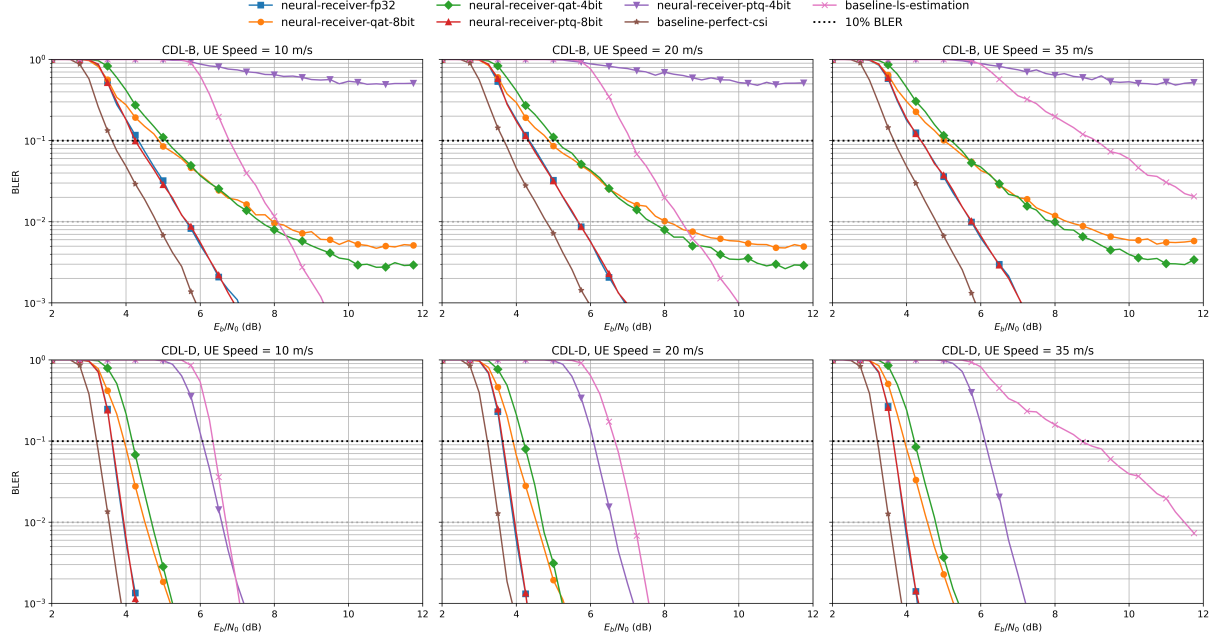
The training objective remains the BCE loss (7), enabling the model to jointly optimize both network weights  $W$  and quantization parameters  $\phi_w$  under the simulated low-precision regime. During each forward pass, all network weights are processed through their respective fake quantization operators, exposing the model to quantization noise patterns it will encounter during actual deployment.

## 4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we evaluate the proposed neural receiver under 4-bit and 8-bit QAT and PTQ. Training is performed on CDL-A, C, and E channels, with validation on CDL-B and D models across the velocity ranges in Table 2, ensuring robust generalization. We report BLER across varying UE velocity, comparing QAT and PTQ receivers against a FP32 neural receiver, Least-Squares (LS)–Linear Minimum Mean Squared Error (LMMSE) equalization with soft demapping, and an ideal receiver with perfect Channel State Information (CSI).

**NLoS comparisons.** The upper panel of Figure 1 illustrates the BLER performance under NLoS conditions, using the CDL-B channel model. The 8-bit QAT configuration reaches 10% BLER within an Signal-to-Noise-Ratio (SNR) range of 4.88–5.01 dB across mobility scenarios, respectively, outperforming the LS baseline by approximately 1.90–4.09 dB as mobility increases. At velocities ranging from 10 to 35 m/s, QAT 4-bit requires 5.09–5.20 dB to attain 10% BLER, maintaining a performance advantage over LS of approximately 1.71–3.90 dB, depending on the mobility level. Across 10–35 m/s, QAT-8bit requires approximately 0.52–0.65 dB higher SNR than FP32 and PTQ-8bit at 10% BLER, whereas QAT-4bit incurs roughly 0.73–0.80 dB over FP32 and 0.75–0.84 dB over PTQ-8bit, consistently across speeds.

**LoS comparisons.** The lower panel of Figure 1 illustrates the BLER performance under NLoS conditions using the CDL-D channel model. The 8-bit QAT configuration achieves 10% BLER within an SNR range of 3.94–3.97 dB across mobility scenarios, respectively, outperforming



**Fig. 1:** BLER vs.  $E_b/N_0$  for the proposed neural receiver under CDL-B and CDL-D channels at various UE speeds.

the LS baseline by approximately 2.4–4.8 dB as mobility increases. At velocities ranging from 10 to 35 m/s, QAT 4-bit requires 4.20–4.23 dB to attain 10% BLER, maintaining a performance advantage over LS of approximately 2.19–4.50 dB, depending on the mobility level. Results show that quantization-aware training is essential for reliable performance at ultra-low bitwidths, as the 4-bit QAT neural receiver consistently outperforms the PTQ variant across all UE speeds.

**QAT 8-bit vs QAT 4-bit.** The notably close performance between QAT 8-bit and QAT 4-bit, differing by only 0.18–0.27 dB in SNR across channel conditions, demonstrates the effectiveness of QAT in learning robust low-precision representations. QAT trains the model to be inherently resilient to quantization noise, reducing sensitivity to precision reduction from 8-bit to 4-bit. Additionally, during QAT, the model learns weight distributions with fewer outliers and better-clustered values, facilitating aggressive quantization for neural network receivers.

**QAT 4-bit vs PTQ 4-bit.** A clear SNR gap emerges between QAT 4-bit and PTQ 4-bit ( $\sim 2$ –3 dB at 10% BLER under LoS, with PTQ failing to reach 10% BLER in NLoS), underscoring the limitations of PTQ at ultra-low bit-widths. QAT adapts weight values during training to optimize for 4-bit representation, whereas PTQ relies on pre-trained weights optimized for full-precision operation. In deep networks like the neural receiver, quantization errors accumulate across layers, which QAT mitigates through training, unlike PTQ, which cannot compensate for inter-layer error propagation.

Interestingly, at 8-bit precision, the PTQ and full-precision

FP32 receivers exhibit a small SNR advantage of  $\sim 0.3$  dB over the QAT receiver. We attribute this to learned weight clipping in QAT, which may under-represent extreme weights compared to PTQs’s calibration-free analytical rounding. Additionally, the STE’s gradient bias and weight-grid oscillations during QAT hinder convergence, leaving this residual gap that could affect radio performance.

## 5. CONCLUSION

We developed and evaluated quantization-aware neural receivers, performing a detailed comparison of QAT and PTQ across diverse channel profiles. We used learned weight clipping via the STE and compared QAT- and PTQ-based neural receivers. The 4-bit QAT neural receiver achieves promising performance at 10% BLER in both LoS and NLoS scenarios, in sharp contrast to the poor performance of 4-bit PTQ. It maintains at most a 0.8 dB SNR gap compared to full-precision receivers, demonstrating robustness to aggressive quantization. By enabling up to an  $8\times$  model size reduction over FP32, 4-bit quantization supports lightweight inference on resource-constrained devices, reducing memory requirements and inference latency while directly lowering telecommunication capital expenditures through reduced hardware and deployment costs. The strong LoS performance of 4-bit QAT receivers also positions them as viable candidates for integrated sensing and communication as well as site-specific deployments in 6G.

## 6. REFERENCES

- [1] Mikko Honkala, Dani Korpi, and Janne M. J. Huttunen, “DeepRX: Fully convolutional deep learning receiver,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3925–3940, 2021.
- [2] Fayçal Ait Aoudia and Jakob Hoydis, “End-to-end learning for ofdm: From neural receivers to pilotless communication,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1049–1063, 2022.
- [3] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Blankevoort Tijmen, “A white paper on neural network quantization,” *arXiv preprint arXiv:2106.08295*, 2021.
- [4] Hao Wu, Patrick Judd, Xiaoxia Zhang, Mikhail Isaev, and Paul Micikevicius, “Integer quantization for deep learning inference: Principles and empirical evaluation,” *arXiv preprint arXiv:2004.09602*, 2020.
- [5] Raghuraman Krishnamoorthi, “Quantizing deep convolutional networks for efficient inference: A whitepaper,” *arXiv preprint arXiv:1806.08342*, 2018.
- [6] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” *arXiv preprint arXiv:1712.05877*, 2017.
- [7] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak, “LSQ+: Improving low-bit quantization through learnable offsets and better initialization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.
- [8] Song Han, “Lectures on tinyML and efficient deep learning computing,” <https://efficientml.ai>, 2024, Fall 2024.
- [9] Jakob Hoydis, Fayçal Ait Aoudia, Alvaro Valcarce, and Harish Viswanathan, “Toward a 6G AI-native air interface,” *IEEE Communications Magazine*, vol. 59, no. 5, pp. 76–81, 2021.
- [10] SaiKrishna Saketh Yellapragada, Esa Ollila, and Mario Costa, “Efficient deep neural receiver with post-training quantization,” *arXiv:2508.06275, Accepted for IEEE 59th Asilomar Conference on Signals, Systems, and Computers, October*, 2025.
- [11] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling, “Data-free quantization through weight equalization and bias correction,” *arXiv preprint arXiv:1906.04721*, 2019.
- [12] Andrey Kuzmin, Markus Nagel, Mart van Baalen, Arash Behboodi, and Tijmen Blankevoort, “Pruning vs quantization: Which is better?,” *Advances in neural information processing systems*, vol. 36, pp. 62414–62427, 2023.
- [13] Mart van Baalen, Benjamin Kahne, Emma Mahurin, Alexander Kuzmin, Anna Skliar, and Markus Nagel, “Simulated quantization, real power savings,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2022, pp. 2756–2760.
- [14] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Merlin Nimier-David, Lorenzo Maggi, Guillermo Marcus, Avinash Vem, and Alexander Keller, “Sionna,” 2022, <https://nvlabs.github.io/sionna/>.
- [15] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv preprint arXiv:1308.3432*, 2013.
- [16] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley J. Osher, Yingyong Qi, and Jack Xin, “Understanding straight-through estimator in training activation quantized neural nets,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [17] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz,” Tech. Rep. TR 38.901, 3rd Generation Partnership Project (3GPP), 2020, version 16.1.0.
- [18] K. Pavan Srinath and Jakob Hoydis, “Bit-metric decoding rate in multi-user MIMO systems: Theory,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7961–7974, 2023.
- [19] Mansoor Shafi, Erik G. Larsson, Xingqin Lin, Dorin Panaitopol, Stefan Parkvall, Flavien Ronteix-Jacquet, and Antti Toskala, “Industrial viewpoints on RAN technologies for 6G,” <https://arxiv.org/pdf/2508.08225>, 2025.
- [20] Kaiming He and Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” *Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.
- [21] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.