Deep-Learned Compression for Radio-Frequency Signal Classification

Armani Rodriguez, Yagna Kaasaragadda, Silvija Kokalj-Filipovic*
Rowan University
*rodrig52@students.rowan.edu, kaasar57@students.rowan.edu, kokaljfilipovic@rowan.edu**

***Abstract*—Next-generation cellular concepts rely on the processing of large quantities of radio-frequency (RF) samples. This includes Radio Access Networks (RAN) connecting the cellular front-end and its framework for the AI processing of spectrum-related data, as well as the AI-native air interface. The RF data collected by the dense RAN radio units and spectrum sensors may need to be jointly processed for intelligent decision making. Moving large amounts of data to AI agents may result in significant bandwidth and latency costs. We propose a deep learned compression (DLC) model, *HQARF*, based on learned vector quantization (VQ), to compress the complex-valued samples of RF signals comprised of 6 modulation classes. We are assessing the effects of HQARF on the performance of an AI model trained to infer the modulation class of the RF signal. Compression of narrow-band RF samples for the training and off-the-site inference will allow not only for an efficient use of the bandwidth and storage for non-real-time analytics, and a decreased delay in real-time applications, but also for efficient AI models in the air interface. While exploring the effectiveness of the HQARF signal reconstructions in modulation classification tasks, we highlight the DLC optimization space and some open problems related to the training of the VQ embedded in HQARF.**

I. Introduction

Data reconstruction from lossy compression incurs a loss of information when the information rate in bits drops below the theoretical lossless minimum, equivalent to the data entropy [1]. If a trained model is used to infer data properties from the reconstructions that suffered information loss relative to the training data, its performance may deteriorate [2], [3]. This paper considers digitally-modulated radio-signal samples in the baseband, intended for the use by a remote deep learning (DL) model trained to infer the signal modulation from such samples. Next-generation (NextG) cellular concepts will rely on the processing of large quantities of RF samples. This includes the new Radio Access Networks (RAN) integrating the cellular front-end with the multi-access edge computing (MEC) architecture and the RAN Intelligent Controller (RIC) framework for AI/ML processing of the spectrum-related data. The RF data collected by the RAN radio units of multiple adjacent NextG cells and spectrum sensors may need to be jointly processed for intelligent decision-making. This may happen both at the edge and in the cloud. Meanwhile, the AI-native air-interface is being addressed by the standard bodies [4], including the data compression for raw data. Moving large amounts of data results in significant bandwidth and latency costs [5]. We believe that it is important to explore the possibility of RF data compression that would preserve the utility of the data. We here apply DL compression (DLC, or learned compression - LC) to compress the complex-valued samples of RF signals comprised of 6 modulations classes. We are assessing the effects of such compression on the performance of an AI model trained to infer the modulation class of captured RF signals and then make intelligent decisions based on their classification.

Machine-learning-based modulation classification, known as ModRec, is an important part of the RF machine learning (RFML) [6] used in spectrum management, interference detection and threat analysis. While exploring the feasibility of the compression of baseband RF samples for the RFML training and off-site inference by the modulation classification task, this paper also highlights some open problems related to vector-quantization methods embedded in the LC training. In this setup, an RF datapoint, which is an array of complex-valued narrowband RF samples, is to be reconstructed from its compressed representation by the user of the ModRec classification model. The compressed representation will be received over a network or retrieved from a storage with no errors. Note the presence of two RFML models, one for the *learned compression* (LC) that we propose here, and another for the ModRec which evaluates it.

Prior and Proposed LC work: An LC model is trained to seamlessly compress data using DL algorithms. LC may leverage discriminative models such as autoencoders [7], or generative models such as variational autoencoders (VAE) [8] and generative adversarial networks (GAN) [9]. The most popular LC architectures typically include a neural net backbone built upon the VAE architecture [10]. One of the latest deep compression models, known as VQ-VAE [11], is an extension to VAE that employs learned vector quantization (VQ). For 30 years, since [12], the learning of optimal VQ codebooks has been an open problem resulting in many attempts to generate a converging algorithm that could learn the quantization vectors for any type of data. The LC proposed here, *Hierarchical Quantized Autoencoder for RF data (HQARF)*, will be analyzed using a family of models, starting from a hierarchical autoencoder, trained using only the reconstruction loss, via an extended model that performs vector quantization (VQ) of the autoencoder's latent space, and ending with a generative model, like VQ-VAE, whose generative loss compares the posterior of the quantized latent representation with a categorical prior. The generative model is motivated by the possibility to leverage statistical diversity of reconstructions [13] to mitigate reconstruction loss or adversarial attacks [14], [15], [16]. The trainable VQ codebook [12], [17] helps to achieve a desired compression rate while maximizing the task-based utility of the reconstructions. To allow for scalability, HQARF maintains a hierarchical architecture. This hierarchical architecture is based on [18] in which a hierarchical version of VQ-VAE,

called Hierarchical Quantized Autoencoder (HQA) has been applied to simple image datasets. To the best of our knowledge, learned compression has not yet been applied to the RF data. We will therefore first explore it in a small, task-specific context, aiming to assess how lossily compressed RF data affects the accuracy of a modulation classification model whose training dataset did not include lossy compression. We motivate the problem and define the basic model in Sec. II. We discuss the details of the compression architecture and the training process, including the loss functions in Sec. III. The classification model and the evaluation of the HQARF reconstructions are discussed in Sec. IV. We conclude in Sec. V.

II. System Model

The hierarchical nature of HQARF allows us to use the same compression model adaptively for different compression rates, and analyze the effectiveness of the quantization on different levels. Multiple compression rates may be important for joint network source coding. Fig.1 depicts our system model where after the compression is done by HQARF, the compressed representation from the desired level (or multiple levels) is sent to a ModRec classifier (or stored, awaiting retrieval by the classifier). The reconstruction is performed at the remote site using the same trained HQARF to recover the original data before classifying it by the ModRec model. The reconstruction uses as many hierarchy layers as the compression has used. Our HQARF model made 2 significant modifications to HQA. **First**, we modified the architecture to work with vectors of complex-valued RF samples instead of images, and modified the reconstruction loss to account for the complex phase reconstruction. **Secondly**, we took a hierarchical approach to training and analyzing the model; first, we train a hierarchical autoencoder (HAE), then we transfer-learn a vector-quantized version of that model using the trained weights of the HAE, while adding a trainable VQ codebook which quantizes the HAE bottleneck, accompanied by a loss component that measures the quantization error; finally, we add the generative loss component based on the Kullback-Leibler divergence, effectively creating a hierarchical VQ-VAE for the RF data (HQARF).

III. HQARF Compression of RF Datapoints

Lossless compression is about finding the shortest digital representation (in bits) of a signal. Lossless compression algorithms take as an input arbitrary information signal represented (sampled) as a sequence of N symbols and process it with the objective to find its shortest compressed representation: a sequence of uniformly distributed bits which cannot be compressed further without a loss of information. Consequently, lossless compressed representation allows for complete reconstruction of the original sequence of N symbols. Information theory sets the foundations for entropy coding, with the length n of the shortest lossless representation equal to the signal's entropy, resulting in a rate $R = n/N$ bits-per-symbol. *Lossy compression* achieves an even shorter representation and lower rate R but the signal reconstructed from such representation suffers a distortion D from the original signal. However, certain utilities of the signal reconstructed from the lossy compression may be unaffected. For example, lossy reconstructed data with a distortion D_u may still be fully classifiable by a deep learning model that was trained on uncompressed data.

Using HQARF to generate lossy reconstructions, we will analyze the effect on the classification accuracy depending on the compression rate $r_\bullet$ (the size in bits relative to the original size). We will compare these with the original of the unit compression ratio. Here, different compression rates r_i are expected to match different bandwidths under a low-latency transmission τ_{RF}, and/ or different storage capacity. Please see Fig 1 where the original x requires bandwidth $\geq B^*$ to be transmitted to the remote classifier within latency τ_{RF} while the HQARF hierarchy levels $i \in \{0, \cdots 4\}$ compress x to fit the bandwidth $B^i < B^*$. To be able to assess the feasibility of a given classification accuracy under the constraint B^i, we next explain the dataset, its compression model and the methodology of its training in detail.

A. HQARF Dataset for modulation recognition (ModRec)

Consider the problem of inferring a property of a physical signal from the signal's reconstruction $\hat{x}$ out of a lossily-compressed representation. We narrow down that question to inferring the modulation class and assessing the classification accuracy in a deep learning setup, given the compression rate. We denote by $MR_\theta(x)$ the algorithm for modulation classification (*ModRec*), where the weights θ have been trained on $\{x \in X\}$. We are interested in comparing $A(\hat{x})$ and $A(x)$, where $A(\bullet)$ is the accuracy of MR_θ. The datapoint x can be represented as $x = [Re_i + jIm_i], i = 1 \cdots p$ with $j = \sqrt{-1}$. We next describe how x is created from a modulated RF signal u, obtained as $u = M_s(b)$, where $s \in \mathcal{S}$ is the employed modulation scheme, and b are information bits. S denotes the finite set of available digital modulation schemes. In this work,

$$\mathcal{S} = [4ask, 8pam, 16psk, 32qam - cross, 2fsk, ofdm256],$$

so we refer to our dataset as *6Mod*. $M_s = \{0,1\}_m \rightarrow \mathcal{C}_n$ describes the modulation function. The sequence of bits $b = \{0,1\}_m$ is encoded into a sequence of complex valued numbers of length n, where the complex sample $c_i = Re_i + jIm_i$, encodes the modulation phase $\phi = \arctan Re_i/Im_i$, and amplitude $a_i = \sqrt{Re_i^2 + Im_i^2}$. We create datapoints as subsequences x of $u \in \mathcal{C}_n$, of length $p = 1024$. Depending on the modulation, x contains more or less mappings of the original random sequence of bits b. This leads to an imbalanced approach to classifying modulations because in any given sequence of length p we will see more randomness due to the original random bit sequence b in low-order modulations than in high-order modulations. Additionally, there is a problem of mistaking one modulation with another whose phase constellation is a subset of its own (like *4ASK (1)* is of *8PAM (2)*, and *16PSK (3)* is of *32qam-cross (4)*), which we illustrate in [19]. However, as this is independent of HQARF, we do not consider its effects on the classification accuracy.

We prepared a synthetic modulation dataset by using the open-source library *torchsig* featured in [20]. The torchsig

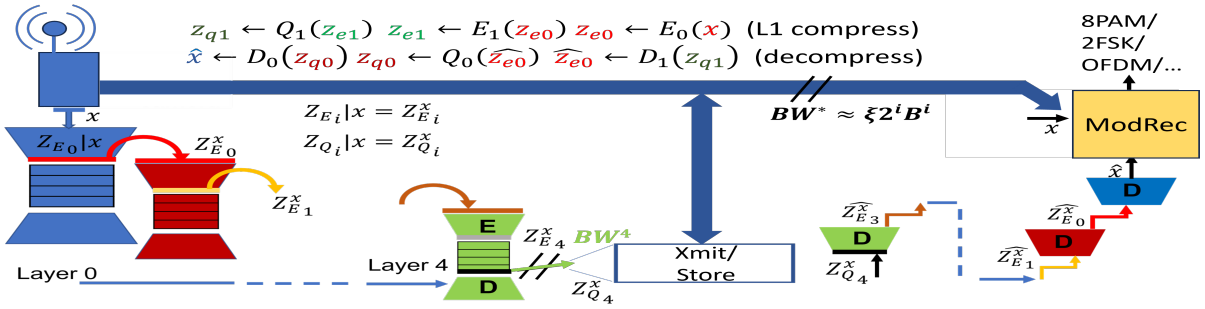

Fig. 1. The RF data x from a SDR goes through HQARF compression layers $i \in \{0, \cdots, 4\}$ requiring bandwidth B^i, to store or transmit the compressed information, vs. directly storing/transmitting x and consuming B^* for a remote classifier to infer its modulation class. x is composed of 1024 complex-valued samples. If a compressed representation Z_{Q_i}, is stored/ transmitted, the same HQARF model is used to recover x, decompressing Z_{Q_i} into $\hat{x}$. We want the ModRec not to see a difference between $\hat{x}$ and x for all $i \in \{0, \cdots, 4\}$. $\xi = 1.37$.

library here emulates the clear-channel samples of high SNR while the effect of the channel and receiver imperfections will be addressed in future research, by leveraging the natural denoising properties of autoencoders in the process of training. The library function *ComplexTo2D* is used to transform vectors of complex-valued numbers into the the 2-channel datapoints, with each channel comprised of p real numbers, previously normalized. Channel 1 contains the real components *(I)* and channel 2 the imaginary ones *(Q)*. Datapoints that are 2-D real-valued vectors required modifications of the architecture in [18] (see Section III-C).

B. Generative DLC with hierarchical VQ-VAE

Architecturally, a VQ-VAE is composed of 3 modules (Fig. 2): **E** - the Encoder (with output z_e), **Q** - the Vector-Quantizer (with output z_q) and **D** - the Decoder which produces the reconstruction of the input x, denoted $\hat{x}$. The HQARF uses a hierarchy of VQ-VAEs in which the encoder's output of the first layer (L0) z_e (creating the least compressed reconstruction) is the input into the second VQ-VAE and so on (Fig. 1). The ith z_e is of dimension $dim(z_e^i) = (\ell, p/2^{i+1})$. The VQ-VAE model projects z_e into discrete latent space z_q as illustrated in Fig. 2. The latent representation z_q produces lower information rate I_{z_q}. The hierarchy of the Encoder-Decoder (E-D) blocks (representing an autoencoder - AE), which is of the same architecture as the respective HQARF blocks, but trained without the Q block and a generative loss, is denoted here as HAE. We will refer to the outermost level of HQARF as VQAE0 and to the same architecture without the **Q** block as AE0. The output z_e of the E block, is of the same dimension in VQAE0 and AE0. Note that *it does not have to be lower than the input's dimension* for the compression to happen. The compression is achieved by adding the Q block with output z_q. In fact, the z_e in the VQAE0 (AE0) of the HQARF showcased here projects the input x of dimensions 2×1024 into z_e of dimensions $\ell \times z_{e_n}$, where $dim(z_e)[0] = \ell = 64$, and $dim(z_e)[1] = z_{e_n} = 512$. Obviously, AE0 does not act as a compression model, but the VQAE0 does. Due to the complexity of training all 3 components (E+D+Q) simultaneously, we performed the following ablation study. We first train the HAE, using the reconstruction loss $L_R(x, \hat{x})$, and then transfer its learned weights to the respective blocks of the HQARF. Next, we train HQARF

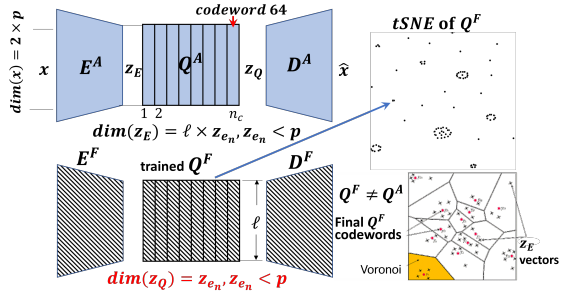

Fig. 2. Training VQ-VAE - **top:** randomly initialized parameters of the encoder (E), decoder (D) and the n_c=16 quantization codebook (Q) codewords of dimension ℓ; **bottom:** the final trained VQ-VAE where a single codeword's index from the trained Q (Q^F) will be associated with each of the $z_{e_n} = dim(z_e)[1]$ slices of x's latent projection z_e. x is compressed into z_q of $z_{e_n} \times \log_2(n_c)$ bits. **Top right:** The t-SNE visualization of Q^F shows clusterization around a few codewords, illustrated with unequal Voronoy cells in the bottom right.

using a modified loss including the additional component which measures the quantization error, the commitment loss $L_Q = E_{q(z_q=k|x)}\|z_e(x) - e_k\|^2$, where e_k is the codeword k of the quantization codebook (see (3) for the definition of the posterior $q(z_q = k|x)$). Note that every hierarchy layer trains a separate E, Q and D block. Finally, after training this hierarchical vector-quantized HAE, we add a generative loss function and retrain HQARF to its final version. The generative loss is a Kullback-Leibler (KL) divergence between the posterior $q(z_q = k|x)$ and the categorical prior with n_c classes, where n_c is the number of codewords in Q.

The n_c codewords (vectors) $e_j, j \in 1, \cdots, n_c$ are of dimension ℓ. Hence, for VQAE0's z_e^0, each one of its $z_{e_n} = 512$ slices of dimension $\ell = 64$ will be represented by a number, indexing a single codeword e_j out of the $n_c = 64$ codewords. For each of 512 slices, the reconstructing user receives this index, losslessly represented by $\log_2(n_c)$ bits. Information rate of the compressed representation z_q can be calculated as $I_{z_q} = z_{e_n} \times \log_2(n_c)$,where $z_{e_n} = dim(z_e)[1]$ (Fig. 2). This is possible as we are parameterizing the E architecture by the tuple (ℓ, h), to yield the dimensionality of the *latent slice* $dim(z_e)[0] = \ell$, making it part of the ℓ-dimensional space in which the Q resides (see the Voronoi tessellation in Fig. 2). We explain the effect of h in Sect. III-C.

The z_q in other HQARF layers will be quantized similarly

as we make sure by the architecture design that each layer's $dim(z_e)[0] = \ell$, equal to the codeword length. Although the codebooks across layers are of the same size 64×64, they do not have to be, given that the VQ-VAEs are independently trained. As far as their training is concerned, the codebooks are agnostic of where the training data is coming from. The optimal Q dimension is an open question.

As x consists of $2p$ real numbers, organized in 2 channels [20] and normalized, we consider each element of x to be independently drawn from a normal Gaussian distribution. It is known that Gaussian has the largest entropy $H_N(X)$ of all distributions of equal variance. For unit variance, $H_N(X) = 1/2 \log(2\pi e) = 2.05$. Hence, the information rate, expressed in the number of bits for each input x is $I_x = dim(x)H_N(x)$. Recall that z_q, the quantized version of the bottleneck z_e, follows multivariate categorical distribution of size n_c, as each of $z_e[1]$ slices of z_e will be represented by the index of one of the codewords e_j. Hence, z_q's dimension is just $dim(z_e)[1] = z_{e_n}$, and each of the z_{e_n} elements is described by $\log_2(n_c)$ bits. For $n_c = 2^d$, the *compression ratio* will be

$$CR = \frac{I_x}{I_{z_q}} = \frac{dim(x)H_N(X)}{z_{e_n}\log_2(n_c)} = \frac{2.05 dim(x)}{d \times z_{e_n}}. \quad (1)$$

If d is such that $d < 2.05$, it leads to $CR \geq 1$ for each z_e with $dim(x) \geq z_{e_n})$, meaning that z_q compresses such x. We want to allow for a larger codebook to be able to perform good vector quantization training: if instead of $n_c = 2^{2.05} \approx 4$, we use $n_c = 2^6$ $(d = 6)$, we must design z_{e_n} to be significantly lower than $dim(x)$ in order to achieve sufficient compression. Under this premise, we design the architecture of the $E - D$ on each hierarchy level to give us $z_{e_n} = dim(input)[1]/2$. Here, *input* is the input to that specific hierarchy level. Hence, for L0, we have $CR_0 = \frac{2.05 \times 2p}{6p/2} = 1.37$, as $dim(input) = dim(x) = 2p$, and $z_{e_{n(0)}} = 512$. The codeword index per slice of z_e is all that we transmit (store) on any compression level, given the user's knowledge of the trained codebooks. For any other level $i > 0$, $CR_i = CR_{i-1} * \frac{I_{z_{e(i-1)}}}{I_{z_{q_i}}}$ and the input $z_{e_{(i-1)}}$ has the same number of channels ℓ as the bottleneck $_{q_i}$, hence,

$$CR_i = \frac{I_x}{I_{z_{q_i}}} = \frac{B^*}{B^i} = \frac{2pH_N(X)}{d \times dim(z_{e(i)})[1]} = \frac{4.1p}{6p/2^{i+1}} = \xi 2^i, \quad (2)$$

where $\xi = 1.37$ is featured in Fig. 1. These calculations yield the *compression rate* $r_0 = 1/CR_0 = 0.73$. Each subsequent reconstruction's dimension is decreased by 2, resulting in $r_4 = 0.73/16 \approx 0.045$. On the other hand, our approximation $I_x = dim(x)H_N(x)$ is very conservative because the **real compression gain is much bigger**. CR_i in (2) may be considered a lower bound, as in practice each of the $2p$ components of x is represented in the single precision floating-point format, consisting of 32 bits, rather than 2.05.

C. Neural Net architecture of VQ-VAE in HQARF

The encoder architecture for $z_{e_0} = p/2$ is parameterized by variables ℓ and h, and composed of 3 1-D convolutional layers. The decoder consists of an equal number of 1-D deconvolutions. Despite the simple architecture of the E-D, difficulties in training were caused by the complexity of the E-Q-D hierarchy, the diverse structure of the loss and its stochastic component, and the intricate data structure. To mitigate this, we introduced a novel process of first training a hierarchy of autoencoders, using a 2-component reconstruction loss, and then transfer-learning the hierarchy of VQ-VAEs by transferring the weights of the autoencoders. The bottleneck z_e is the output of third 1-D convolutional layer in E, with ℓ output channels. The other convolutional layers have the number of output channels affected by the parameter h, which is how the learning capacity (number of weights) is controlled across the layers. We started with the parameter values inherited from [18] and concluded that these parameters are not optimal. Our criterion for optimality is based on the comparison of the evaluated classification accuracy $A_i(\hat{x})$ of the L_i reconstructions and the accuracy that we expect based on the singular value decomposition (SVD) that we performed on the original data.

Layer	L0	L1	L2	L3	L4	L5	L6
Input dim	2x1024	64x512	64x256	64x128	64x64	64x32	64x16
E Outp. (z_e_i)	64x512	64x256	64x128	64x64	64x32	64x16	64x8
inp/outp Ratio (HAE)	1/16 **Total:** $^1/_{16}$	2 $^2/_{16}$	2 $^4/_{16}$	2 $^8/_{16}$	2 $^{16}/_{16}$	2 2 (SVD th.)	2 4

Fig. 3. Table of the HAE encoders' input/ output dimensions and the SVD bound (L5) for optimal h parameters.

SVD-based threshold: We performed an SVD on the original 6Mod data in the complex-valued domain, and calculated how many eigenvectors we should keep to preserve more than 99 % of the total information in the data. The result is that we need 500 out of the original 1024 complex eigenvectors. This means that designing z_e s.t. the product of its dimensions is equal to $0.5\,(dim(x))$, allows for the $\hat{x}$ reconstructed from such a z_e to be perfectly classified. Hence, according to the table in Fig. 3, L5 is our *SVD bound*, as its z_e has 1/2 of the original dimensions: if we manage to achieve the accuracy $A_5 = 100\%$ by modifying h, it means that the HAE autoencoders are parameterized well (and so are the HQARF's). With original parameters, A_4 was as low as 50% while it is 80% in Fig. 5 and above 90% in Fig. 6, almost closing the gap to the SVD bound. We are in the process of optimizing the size of the codebook to achieve the best HQARF performance for the new HAE architecture as it experienced a drop despite a better HAE (Fig. 6).

The Q is designed as a learnable tensor of dimension $n_c \times \ell$, s.t. we can train it based on the MSE distance d_{MSE} between each e_k of length ℓ, and each of the z_{e_n} slices of length ℓ. As in [18], we pick the e_k to quantize each slice using a stochastic method, based on sampling the posterior probability

$$q(z_q = k|x) = \exp^{-\|z_e(x) - e_k\|^2}, \; k \in \{1, \cdots, n_c\}. \quad (3)$$

This posterior is the basis of the KL loss, which serves to

make $q(z_q = k|x)$ similar to a categorical prior. Generative reconstructions (illustrated in [19]) are important for the data robustness [13]. The Q codewords (CWs) are being learned starting from random Gaussian samples at the initialization, and converging to a Q that minimizes the loss function, composed not only of the reconstruction loss $L_R(x, \hat{x})$, but also the KL generative loss, and a commitment loss measuring the distance between the z_e and the chosen e_k. Note that, in the outermost layer L0, we added a new component to $L_R(x, \hat{x}) = L_{MSE} + L_\phi$, to measure not only the MSE distance between x and $\hat{x}$, but also the cosine loss

$$L_\phi = 1/p \sum_{i=1}^{p} \frac{x[i,:] \times \hat{x}[i:0]^T}{\|x\| \times \|\hat{x}\|}.$$

As $x[i,:]$ are the real and imaginary parts of the ith RF sample, L_ϕ measures the phase reconstruction, a very important feature in digital phase modulations. For details of the Q training, please consult our code [21]. Apart from the typical tuning of the Q parameters using stochastic gradient descent of the loss, and obtaining a differentiable sample from the posterior (3) via the Gumbel Softmax relaxation [22], the least used CW is periodically reset to the vicinity of the most used CW. We considered the reset period to be a hyper-parameter and obtained good results when it increased, as frequent resetting foster instability. More importantly, we defined the vicinity of the CW adaptively, circling in with the number of resets (see t-SNE [23] visual of the codebook in Fig. 2). The optimal reset policy is actively investigated using the statistics of e_k over the training epochs.

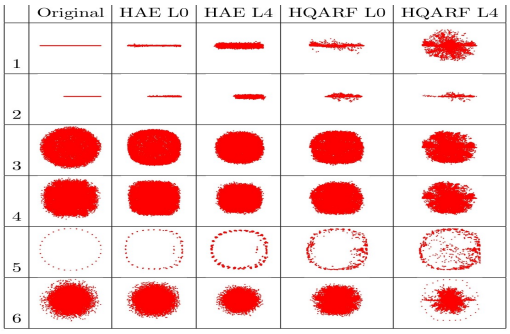

Fig. 4. I/Q scatterplot of 6 different classes based on the reconstructions across layers compared with the ideal (original) scatterplot. We concatenated 20 reconstructions of random datapoints of the same class, each comprised of 1024 complex-valued samples, and plotted them in the complex plane.

IV. Evaluation with the EfficientNet Classifier

Upon training the 5 HQARF Layers on the *6Mod dataset*, we evaluated it on a modified **EfficientNet_B4** [24] referenced in [20], which was appropriately transfer-learned on the original *6Mod* dataset. Evaluation gave us a reference accuracy $A(x) \approx 100\%$. Fig. 5 shows how the accuracy of reconstructions depends on the compression ratio (CR). The HAE accuracy should not be associated with the CR in the x-axis. It is there to illustrate the SVD gap, i.e., if the space of the h parameter, and possibly the overall architecture, should be further explored (as emphasized by the accuracies of the 2 trained models with different h in Figs. 5 and 6). Fig. 4 shows "'digital constellations"' of the originals and their reconstructions. While the real constellations show complex samples at symbol times, ours are the scatterplots of complex samples at a much higher rate obtained by baseband sampling. However, they illustrate the gradual deterioration in the phase reconstruction while the ModRec utility follows the trend (Figs. 5 and 6).

V. Conclusions and Future Work

We introduce HQARF, the first vector-quantization (VQ) based learned compression (LC) of modulated RF signals and evaluate their lossy reconstructions on a modulation recognition (ModRec) task, illustrating the utility of LC in this domain and its optimization space. Based on our results, this proof of concept deserves further investigation, as it may have applications in intelligent network optimization where large quantities of RF samples need to be collected to train the AI in NextG cellular algorithms [4]. Moreover, vector-quantized latent representations of RF signals can be useful in the design of the diffusion-based AI-native air-interface, such as [25], where quantizing the latent space would help achieve a better trade-off between quality and speed [26]. The simple architecture and compact size of HQARF are very convenient for the quantization close to the radio interface. We point out to the complex factors affecting the ModRec accuracy on the HQARF reconstructions, but also the fidelity of their complex-plane scatterplots and spectrograms (which is the focus of a companion paper [19]). These optimization factors include the HQARF architecture, training methodology, loss functions and the dimension and training of the VQ codebook. We defined a bound for the LC performance based on SVD. Pursuing this bound by tuning optimization factors, we kept improving our results, and plan to continue doing so in the future.

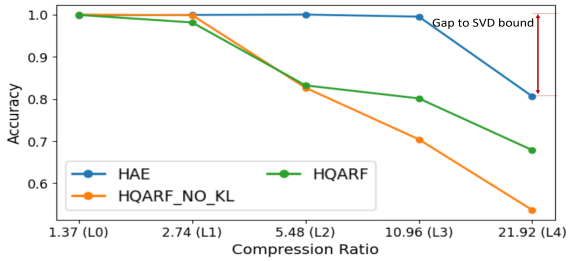

Fig. 5. Accuracy vs compression ratio (CR) across Layers for HAE, HQARF_NO_KL and HQARF with Q of size 64×64. The CR on the x axis **does not apply to HAE**, as HAE does not perform VQ: HAE is added to track how close we are to the bound given by SVD.

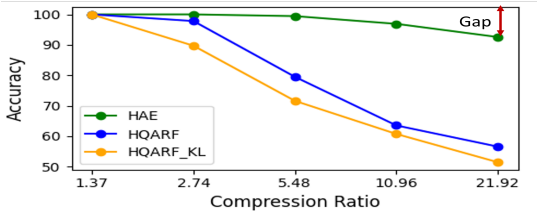

Fig. 6. Improved HAE architecture almost closes the gap to the SVD bound, but the HQARF performance experiences a drop. We will continue to optimize the codebook size to leverage the HAE gain in the HQARF performance.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal*, vol. 27, 1948.

[2] F. Codevilla, J.-G. Simard, R. Goroshin, and C. Pal, "Learned image compression for machine perception," *ArXiv*, vol. abs/2111.02249, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:241033392

[3] M. Williams, S. Kokalj-Filipovic, and A. Rodriguez, "Analysis of lossy generative data compression for robust remote deep inference," in *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, 2023.

[4] G. R. 18, "Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface RAN," 2023, [Meeting 112, 27th February – 3rd March, Athens, Greece, Technical Report.].

[5] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and Learning in O-RAN for Data-driven NextG Cellular Networks," vol. 59, no. 10, 2021.

[6] S. Peng, S. Sun, and Y.-D. Yao, "A Survey of Modulation Classification Using Deep Learning: Signal Representation and Data Preprocessing," *IEEE trans. on neural networks and learning systems*, vol. 33, no. 12, 2022.

[7] C. Jia, Z. Liu, Y. Wang, S. Ma, and W. Gao, "Layered Image Compression Using Scalable AutoEncoder," in *IEEE Conf. on Multimedia Inform. Processing and Retrieval (MIPR)*, 2019.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ArXiv*, vol. abs/1312.6114, 2013.

[9] F. Mentzer et al., "High-fidelity generative image compression," *ArXiv*, vol. abs/2006.09965, 2020.

[10] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning End-to-End Lossy Image Compression: A Benchmark," vol. 44, no. 8, 2022.

[11] A. V. den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," in *31st Intern. Conf. on Neural Information Processing Systems*, 2017.

[12] T. Kohonen, "LVQ-.PAK Version 3.1," 1995, [LVQ Programming Team of the Helsinki University of Technology].

[13] S. Kokalj-Filipovic and M. Williams, "Generative Lossy Sensor Data Reconstructions for Robust Deep Inference," in *International Balkan Conference on Communications and Networking (BalkanCom)*, 2023.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6572

[15] C. Szegedy et al., "Intriguing properties of neural networks," *arXiv*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6199

[16] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv*, 2015. [Online]. Available: https://arxiv.org/abs/1412.5068

[17] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. on Information Theory*, vol. 44, no. 6, 1998.

[18] W. Williams et al., "Hierarchical quantized autoencoders," in *34th Intern. Conf. on Neural Information Processing Systems (NIPS)*, 2020.

[19] Y. Kaasaragadda, A. Rodriguez, and S. Kokalj-Filipovic, "Can We Learn to Compress RF Signals?" in *IEEE International Balkan Conference on Communications and Networking (BalkanCom)*, 2024.

[20] L. Boegner, M. Gulati, G. Vanhoy, P. Vallance, B. Comar, S. Kokalj-Filipovic, C. Lennon, and R. D. Miller, "Large Scale Radio Frequency Signal Classification," 2022. [Online]. Available: https://arxiv.org/abs/2207.09918

[21] S.Kokalj-Filipovic, A. Rodriguez, and Y. Kaasaragadda, "HQARF code," https://github.com/skokalj/vq_hae_1D/, 2024.

[22] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," in *ICLR*, 2017.

[23] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Mach. Learning Research*, vol. 9, no. 86, 2008.

[24] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Intern. Conference on Machine Learning, (ICML)*, 2019.

[25] M. Letafati, S. Ali, and M. Latva-aho, "Generative AI-Based Probabilistic Constellation Shaping With Diffusion Models," 2023.

[26] Shuyang Gu et al., "Vector quantized diffusion model for text-to-image synthesis," in *IEEE Conf. on Computer Vision and PatternRecognition*, 2022.