Can We Learn to Compress RF Signals?

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

***Abstract*—The AI based on radio spectrum sensing is being deployed to detect and classify various interference sources and optimize spectrum allocation in next-generation cellular networks. Radio frequency (RF) spectrum sensing for AI must generate large quantities of RF data whose transport to the AI in the Cloud and at the Edge will incur significant bandwidth and latency costs. Can these quantities be compressed without affecting the utility of the cellular AI models? Our deep learned compression (DLC) model, named HQARF and based on learned vector quantization (VQ), compresses and reconstructs complex-valued samples of RF signals comprised of different modulation classes. In this paper we analyze how the salient (signal-processing) properties of the RF signal reconstructions by HQARF are modified due to incremental compression, and how this affects the accuracy of an AI model trained to infer the signal's modulation class. This analysis is also important as many advanced AI models developed for vision, such as *Stable/VQ Diffusion*, are trained on the quantized latent representation of the data. Understanding effects of learned RF quantization may help leverage those advanced models in the RF domain.**

I. Introduction

We consider digitally-modulated radio-signal samples in the baseband, intended, without loss of generality, for the use by a remote deep learning (DL) model trained to infer the signal modulation from such samples. In modern cellular networks, the RF data collected by the Radio Access Network (RAN) radio interfaces and collocated spectrum sensors may need to be jointly processed at the edge or in the cloud for intelligent decision-making. As the data transmission to the remote AI will incur significant bandwidth and latency costs, we aim to answer the question if it can be compressed without affecting the utility of the cellular AI models. Our DL compression (DLC) method belongs to a class of algorithms typically called *learned compression* (LC). It compresses complex-valued samples of RF signals comprised of different modulations classes. Using 6 common modulations we aim to assess and illustrate how various salient properties of the RF signal reconstructions by DLC are modified due to incremental compression and relate that to the performance of an AI model trained to infer the signal's modulation class.

Machine-learning-based modulation classification recognition/ (ModRec) has been on the forefront the RF machine learning (RFML) [1], an interdisciplinary approach that combines expertise in RF engineering, signal processing, and machine learning to address challenges in wireless communication systems used in spectrum management, interference detection and threat analysis, and in the AI-native air-interfaces. Compression of the baseband RF samples for the RFML training and off-site inference will allow for an efficient use of the bandwidth and storage for non-real-time analytics. Let us assume here that the compressed representation will be received over a network by the ModRec task, or retrieved from a storage with no errors. Please note that both the ModRec and the *HQARF*, the proposed LC model, are DL models. An LC model is trained to seamlessly compress data using DL algorithms. DLC may leverage discriminative models such as autoencoders [2], or generative models such as variational autoencoder (VAE) [3]. The *HQARF* is analyzed using a family of models, from a hierarchical autoencoder, trained using only the reconstruction loss, via an extended model that performs vector quantization in the autoencoder's latent space, to a generative model which adds a generative loss. The trainable vector quantization [4], [5] helps to achieve a desired compression rate while preserving the task-based utility of the reconstructions. To allow for scalability, HQARF maintains a hierarchical architecture.

To the best of our knowledge, HQARF is the first LC model applied to the RF data, described in more detail in a companion paper [6]. We here study its effects on the salient properties of the RF signal reconstructions, how they change with the incremental compression and how they relate to the performance of an AI modulation classification model whose training dataset did not include lossy compression. We define the basic problem in Sec. II. We discuss the details of the HQARF architecture and the training process, including the achieved compression rates, in Sec. III. We analyze salient properties of the HQARF reconstructions in Sec. IV and conclude in Sec. V.

II. Problem Definition

The DLC architectures are typically based on a neural net backbone built upon the VAE architecture [7]. One of the latest deep compression models, known as Vector-Quantized Variational Autoencoder (VQ-VAE) [8], is an extension to VAE that employs learned vector quantization (VQ). HQARF is one such method, modified from [9], in which a hierarchical version of VQ-VAE, called Hierarchical Quantized Autoencoder (HQA) has been applied to simple image datasets. The hierarchy in HQARF allows us to use the same compression model adaptively for different compression rates, and analyze its effectiveness incrementally. Fig.1 shows 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 remote ModRec (or stored, awaiting retrieval by the classifier). The reconstruction is performed at the remote site using the same trained HQARF to recover

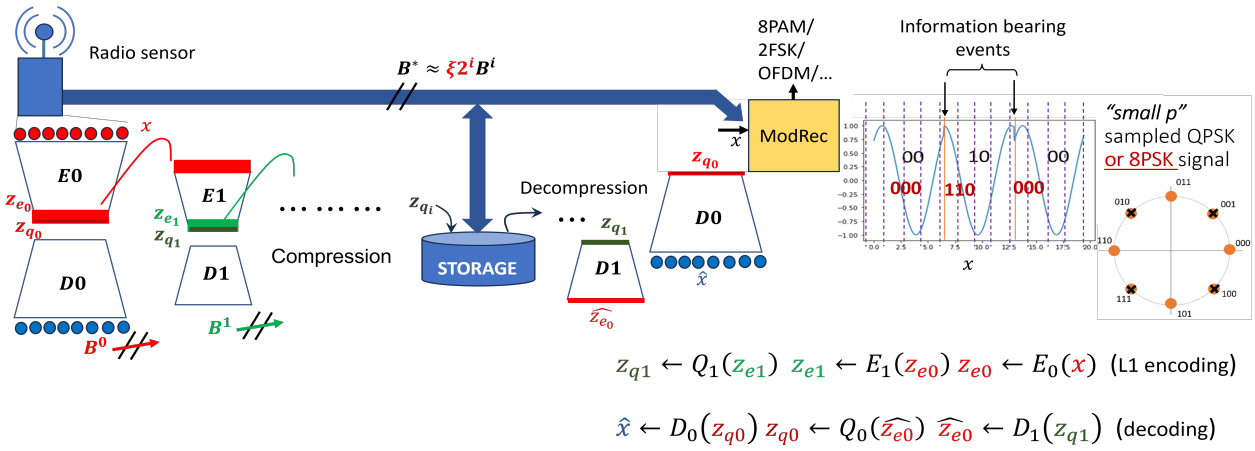

Fig. 1. Information flow from a sensor through HQARF compression layers $i \in \{0, \cdots, 4\}$ demanding bandwidth amount of B^i, to store/ transmit the information about the x composed of $p = 1024$ complex-valued samples, vs. directly storing/transmitting x for a remote ModRec to use. After a compressed representation z_{q_i} is stored/ transmitted, the same HQARF model is used to recover x, decompressing z_{q_i} into $\hat{x}$. We measure the compression effectiveness by comparing the ModRec accuracy on the reconstructed datapoint $\hat{x}$ and the original x for various compression rates B_i/B^*. The sensor does not know symbol boundaries (marked as information bearing events), so it uniformly samples the entire signal (top-right). In the top-right, we also illustrate the *small-p* effect: when the datapoint is short, we may confuse 2 modulations, if one is a subset of another (like QPSK and 8PSK, or in this work the pairs *4ASK (1) and 8PAM (2)*, and *16PSK (3) and 32qam-cross (4)*).

the original data before classifying it by the ModRec. The reconstruction uses as many layers of hierarchy as the compression have used. Note that the data point x is a sampled RF signal (waveform), as illustrated in the top-right corner of Fig.1. Next to the waveform, we also illustrate a problem that we refer to as the *small-p* effect. It happens when due to the insufficient number p of RF samples in x, we confuse one modulation with another, if the constellation of the latter is a superset of the former. Compared to HQA, we made the following modifications: 1. we modified the neural net to work with vectors of complex-valued RF samples instead of images, and augmented the reconstruction loss to include a cosine loss which measures the fidelity of the complex phase reconstruction; 2. we took an incremental approach to training and analyzing the model; 3. we optimized the training of the VQ codebook.

Using HQARF to generate signal reconstructions $\hat{x}$, we will analyze the effect on the classification accuracy depending on a hierarchy of compression rates $r_\bullet$ (the size in bits relative to the original size). We compare these with the original of the unit compression ratio. Here, reconstructions of different compression rates r_i require different bandwidth, and/ or different storage capacity. 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 hierarchical levels $i \in \{0, \cdots 4\}$ compress x to fit the bandwidth $B_i < B^*$.

III. Hierarchical Vector-Quantized Compression of RF Datapoints

This paper studies the signal processing properties of the hierarchical HQARF reconstructions and their usefulness for modulation classification. We first explain the HQARF compression model and the methodology of its training.

A. Dataset for HQARF and ModRec

Let $MR_\theta(x)$ be the ModRec model, whose 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 $j = \sqrt{-1}$. Here, a modulated signal u is obtained as $u = M_s(b)$, where $s \in \mathcal{S}$ is the employed modulation scheme, with S denoting the finite set of available digital modulation schemes. In this paper,

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

so we refer to our dataset as *6Mod*. For any s, $M_s = \{0,1\}_m \rightarrow \mathcal{C}_n$ describes the modulation function with modulation class s. The sequence of bits $b = \{0,1\}_m$ of length 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 sub-sequences x of $u \in \mathcal{C}_n$, of length $p = 1024$, which depending on the modulation contains more or less mappings of the original random sequence of bits b. This leads to an imbalanced approach to classifying modulations, as in any given sequence of length p we will see a larger ratio of the constellation points in low-order modulations than in high-order modulations, leading to misinterpretations (the top right in Fig 1 illustrates possible confusion between QPSK and 8PSK due to partial observability). As this is the common approach in ModRec, we do not consider its effects on the accuracy even though the *6Mod* $\mathcal{S}$ contains diverse modulation orders (how many original bits are represented by a single complex value). Although it is likely that by using $p > 1024$ we may be able to better train both the ModRec and the HQARF, this would require more complex neural net architecture. For simplicity,

we decided to start with a less complex model.

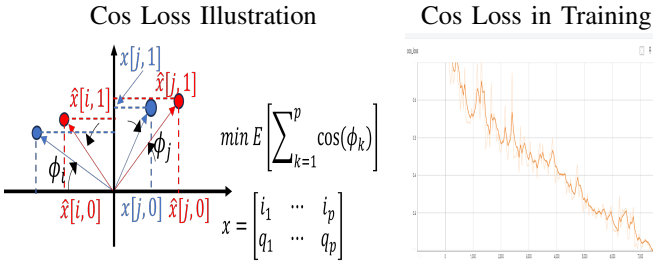

Fig. 2. Minimizing the cosine loss aims to align the reconstructed phase, $arctan(\hat{i}/\hat{q})$, with the original phase $arctan(i/q)$.

We prepared a synthetic 6Mod dataset by using the open-source library *torchsig* featured in [10]. The torchsig library here creates the RF samples of high signal-to-noise ratio (SNR). The library function *ComplexTo2D* is used to transform vectors of complex-valued numbers into the 2-channel datapoints, with each channel comprised of p real numbers, previously normalized. Channel 1 contains real (or in-phase) components *(I)* and channel 2 contains the imaginary (or quadrature) ones *(Q)* (see Fig 2). The fact that our datapoints are 2-D vectors of real numbers required some modification of the architecture in [9] (see Section III-C). One of the major design decisions was to implement HQARF to use 2-D vectors instead of using signal spectrograms which would allow leveraging the abundant open-source code from the image domain and the implementation in [9]. However, spectrogram images lose the information about the modulated phase that cannot be reversed, and lead to suboptimal classification.

B. HQARF: Generative DLC of RF data

We opt for generative DLC in HQARF to investigate the possibility of discrete diffusion [11], [12] in RF domain. The HQARF uses a hierarchy of VQ-VAEs in which the encoder's output of the first layer (L0) z_e (producing the least compressed reconstruction) is the input into the second VQ-VAE and so on (Fig. 1). The layers are numbered $i \in \{0, \cdots, 4\}$ where the ith z_e is of dimension $dim(z_e^i) = (\ell, p/2^{i+1})$. The VQ-VAE model is a generative algorithm that extends the VAE model [13] through the use of a vector-quantized, discrete latent space z_q as shown in Fig. 3. It consists of 3 modules: **E** - the Encoder neural net (with output z_e), **Q** - the Vector-Quantizer (with output z_q) and **D** - the Decoder net which produces $\hat{x}$, the reconstruction of the input x. Quantizing z_e to z_q results in lower information rate $I_{z_q} < I_{z_e}$.

A hierarchy of the Encoder-Decoder (E-D) blocks (as in an autoencoder), 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 refer to the outermost level of HQARF as VQAE0 and 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, and the compression via the Q block projects it into z_q. In this paper, VQAE0 projects the input x of dimensions 2×1024 into z_e of dimensions $dim(z_e)[0] = \ell = 64$, and $dim(z_e)[1] = z_{e0} = 512$.

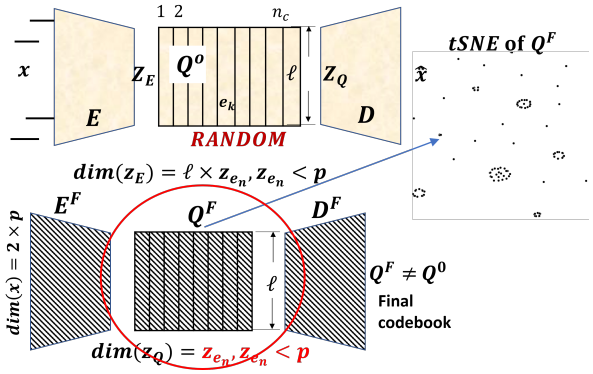

Fig. 3. The training of VQ-VAE - **top:** randomly initialized parameters of the encoder (E), decoder (D) and the n_c quantization codebook (Q) vectors of dimension ℓ; **bottom:** the final trained VQ-VAE where a single codeword's index from the trained Q will be associated with each of the $z_{en} = dim(z_e)[1]$ slices of x's latent projection z_e. Therefore x is compressed to $z_{en} \times \log_2(n_c)$ bits. **On the right:** The t-SNE visualization of the trained Q shows *clusterization* around a few codewords.

Due to the complexity of training all 3 components (E+D+Q) simultaneously, we perform the following transfer-learning. We first train the HAE, using the reconstruction loss $L_R(x, \hat{x})$, and then transfer its weights to the respective blocks of the HQARF. Next, we train HQARF using a modified loss including an additional component which measures the quantization error, the commitment loss

$$L_Q = E_{q(z_q=k|x)} \left\| z_e(x) - e_k \right\|^2,$$

where e_k is the codeword k of the quantization codebook (please see (1) for the definition of the posterior $q(z_q = k|x)$). Every hierarchy layer trains a separate E, Q and D block. Finally, after training this HQARF model, we add a generative loss and retrain HQARF to its final version. The generative loss is a Kullback-Liebler 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.

Given that both x and z_e are composed of real numbers whose values are between -1 and 1, we consider each element to be independently drawn from a normal Gaussian distribution. Hence, the information rate of each dimension is equal to the Gaussian entropy $H_N(X) = 1/2 \log(2\pi e) = 2.05$ for normal distribution of unit variance. This is an approximation given that certain modulations constrain the range of normalized signal amplitudes (e.g., with 4ask and 8pam the Q component is equal to 0). On the other hand, our approximation is extremely conservative, as we could have instead used the actual size of the datapoint equal to $2p \times 32$, since we train the model using single precision floating pointss (of 32 bits).

As each column of the compressed latent z_q is represented by the integer index of a Q codeword, information rate of z_q can be calculated as $I_{z_q} = z_{ei} \times d, d = \log_2(n_c)$. This is because, by design, each of the n_c codewords is of dimension ℓ, same as $dim(z_e)[0]$. For VQAE0, each one of its $z_{e0} = 512$ columns of dimension $\ell = 64$ will be represented by a number 1 to 64 with 6 bits ($d = 6$). Please consult [6] for details how

this leads to the *compression ratio* of layer i :

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

where $\xi = 1.37$, as also featured in Fig. 1. The codeword index per element of the 2nd dimension of z_e is all that we transmit (store), given the user's knowledge of the trained codebooks. As *compression rate* r_i is the inverse of the *compression ratio* CR_i, we have $r_0 = 1/CR_0 = 0.73$, $r_i = 0.73/2^i$.

L_i (h)	L0 (16)	L1(16)	L2(32)	L3(64)	L4(128)	L5(128)	L6(128)
HQARF weight count	21,586	92,218	688,202	2,861,930	11,186,346		
Appr. E weight count	2 x 8 K 8 x 16 K 16 x 64	64 x 8 K 8 x 16 K 16 x 64	64 x 16 K 16 x 32 K 32 x 64	64 x 32 K 32 x 64 K 64 x 64	64 x 64 K 64 x 128 K 128 x 64	64 x 64 K 64 x 128 K 128 x 64	64 x 64 K 64 x 128 K 128 x 64
Input dim	2x1024	64x512	64x256	64x128	64x64	64x32	64x16
Outp. (z_e_l)	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. 4. The number of parameters (complexity) per layer vs input/ latent dimensions and the parameter h; here $K = 3$ as we use the kernel size of 3 across the layers, and the SVD threshold (evaluated on L5) is the x to z_e ratio where we still expect full accuracy, i.e. the z_e would still contain all the information content of x, if appropriately trained.

C. Neural Net architecture of VQ-VAE in HQARF

The encoder architecture for $z_{e0} = p/2$ is composed of 3 1-D convolutional layers, and the decoder consists of an equal number of 1-D deconvolutions. The simple architecture of HQARF is very convenient for the quantization close to the radio interface, as the model can be compactly stored (see the number of parameters in Fig. 4). Despite the simple architecture, it is the complexity of the E-Q-D hierarchy, the diverse structure of the loss and its stochastic component, and the intricate data structure that caused difficulties in training. It made us introduce the transfer-learning in 3 stages. The E-D architecture is adaptable despite the constant $dim(z_e)[0] = \ell$ of the bottleneck z_e, which is the output of the 3rd 1-D convolutional layer in E, with ℓ output channels. The other convolutional layers have the number of output channels affected by the HQARF parameter h, and that is how the learning capacity (number of weights) is controlled across the layers (Fig. 4). Our criterion for optimality is based on the comparison of the evaluated classification accuracy $A_i(\hat{x})$ of the L_i reconstructions by HAE, and the accuracy that we expect based on the singular value decomposition (SVD) that we performed on the original data.

SVD-based threshold: We performed an SVD on the 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. This showed that if we compress z_e to have 1/2 of the original dimensions of x, the $\hat{x}$ reconstructed from such a z_e is likely to be perfectly classified. Hence, according to the table in Fig. 4, L5 is our *SVD bound*: if we manage to achieve the HAE L5 accuracy $A_5 = 100\%$, it means that we parameterized the E-D chain in the HAE optimally (and thus in HQARF). Guided by the SVD bound, we modified the h hyperparameters and improved the performace (Fig. 9), while still not closing the gap (as even A_4 is not perfect). We suspect that the reason for that is the reconstructions loss, as discussed in the next session.

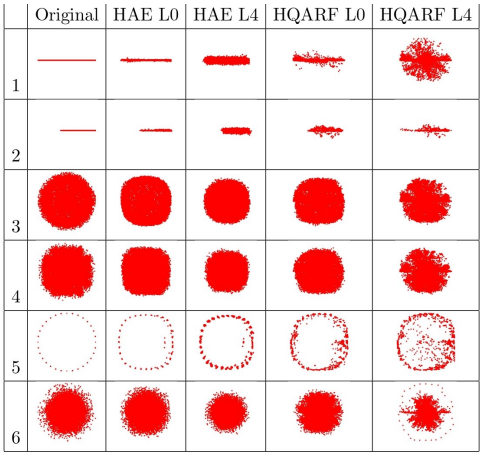

Fig. 5. i/q scatterplot of 6 different classes based on the reconstructions across layers compared with the ideal (original) scatterplot. We concatenated 20 reconstructions of random datapoints of the same class, each comprised of 1024 complex-valued samples, and plotted them in the complex plane.

A stochastic method [9], based on sampling the posterior probability of the codeword e_k, quantizes each slice:

$$q(z_q = k|x) = \exp^{-\|z_e(x) - e_k\|^2}. \quad (1)$$

The learning of the codebook $\{e_j\}, j = 1, \cdots, n_c$ is based on minimizing the global loss function, which contains not only the reconstruction loss $L_R(x, \hat{x})$, but also a generative loss comparing the posterior $q(z_q = k|x)$ with a categorical prior, and a commitment loss that measures the distance between the z_e slices and the chosen e_ks. $L_R(x, \hat{x})$ measures not only the MSE distance between x and $\hat{x}$, but also the cosine loss along their 2nd dimensions, which effectively controls the fidelity of phase reconstruction, a very important feature in digital phase modulations (see Fig. 2).

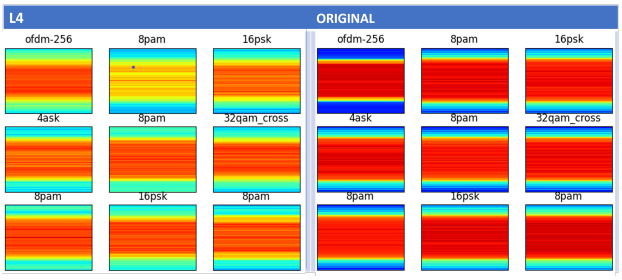

Fig. 6. Spectrograms of datapoints and their L4 reconstructions for randomly selected modulations

IV. Evaluation of the HQARF reconstructions

After training the 5 HQARF Layers on the *6Mod dataset*, we compared the waveform reconstructions with the originals. To illustrate the deficiencies in our approach, we here only show the highest compression reconstructions. By zooming in

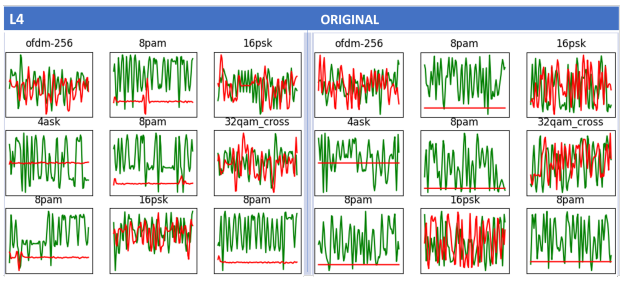

Fig. 7. The first 100 real (green) and imaginary (red) components of L4 reconstructions and their originals for randomly selected modulations

the first 100 samples of both the I and the Q component in Fig. 7, we better visualize the difference between the L4 reconstructions and the originals. The obvious waveform perturbations in the reconstructions are expected for such a low compression rate, because the SVD bound (100 % classification up to layer 5) does not necessarily mean a high-fidelity reconstruction. The information bearing parts of the signal are sparse, hence reconstructing every single sample in the signal does not help classification accuracy.

Besides, small perturbations in the reconstruction arise as we are training a generative model, which not only plays an important part in increasing the reconstruction robustness (which we do not discuss in this paper), but also represents an essential component in the diffusion-based AI models. In the current training for 50 epochs, all 3 components of the loss seem to converge, creating a balanced trade-off between learning to reconstruct and learning the probabilistic model.

We further observe that the average ratio between the I and the Q component is preserved, thanks to the employed cosine loss. However, as the majority of information about the modulation class is located at symbol boundaries, these should be the high-importance samples where both the cosine and the Euclidean distance strongly contribute to the total loss. Hence, our future work will include a loss function that leverages the detection of phase/amplitude shifts based on higher differentials of x to detect likely symbol boundaries. We also intend to use the cosine loss as a regularization term controlled by a hyperparameter that depends on which downstream task is utilizing the reconstructions.

Next, we compared "'digital constellations"' of the original datapoints and their reconstructions (Fig. 5). Note that the real constellations utilize complex samples at symbol times, while we here presented scatterplots of complex samples at a much higher rate (for each datapoint sample). Most of these points are insignificant for the ModRec utility, which goes back to the suggested detection of candidate symbol boundaries (real constellation points) that will be more useful in preserving good classification accuracy, and would for that reason also merit better reconstruction. Despite these drawbacks, Fig. 5 correctly shows gradual deterioration in phase reconstructions with increased compression, as was seen in the time and the frequency domains (although we only presented L4 in Fig. 6 and 7 due to space limitations). Due to the stochastic reconstruction which includes sampling of the posterior, spectrograms of the originals and their L4 reconstructions (Fig. 6) show differences between the reconstructions even within the same class. Another reason for this may be the "small p"' effect (see for example 8PAM in Fig. 6). These stochastic effects exist in the temporal domain too, but they are less tractable there. To track the distortion without a "'small p"' effect, we used spectrogram comparisons given a specific x, like in figure Fig. 8.

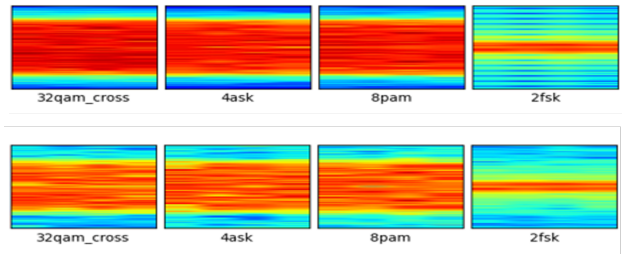

Fig. 8. For specific x (the top row), we observe the relative L4 distortion in $\hat{x}$ (bottom row), which , given x, may also be random, since our compression model is stochastic (1).

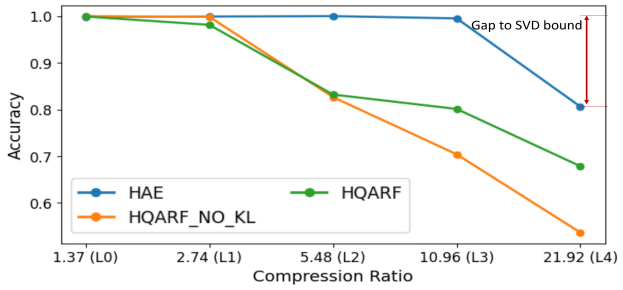

Fig. 9. Accuracy vs compression ratio (CR) across Layers for HAE, HQARF_NO_KL and HQARF with Q of size 64×64. The CR on the x axis **does not apply to HAE**, as HAE does not perform VQ: HAE is added because by tracking how close we are to the bound given by SVD, we know that we can do better than this result.

To complete the assessment, we evaluated our reconstructions on the version of the **EfficientNet_B4** [14] used in [10], which was appropriately transfer-learned, and evaluated on the original *6Mod* dataset, resulting in a reference accuracy $A(x) \approx 100\%$. Note that we evaluated HAE to be able to assess how well the architecture of the encoder and decoder is parametrized (see Fig. 4 and Fig. 9). Fig. 9 shows how the accuracy of the reconstructed data depends on the compression ratio (CR). Note that the HAE does not exhibit such CR. It is included in the plot to illustrate the SVD performance gap. The space of the h parameter and the loss functions should be further explored to close that gap. Fig. 10 shows how the accuracy of the reconstructed data depends on the modulation class, and what modulations classes are less resilient to distortion due to compression. Both Fig. 10 and Fig. 5 show that "'16psk"' (3) and "'32qam-cross"' (4) look alike even for L0, for reasons illustrated in Fig1 due to a small p.

With more compression all classes start looking like "'ofdm256"' (6). This is because OFDM signals look like sharply band-limited white noise to the non-collaborative

receiver such as an RF sensor, so the more noisy other reconstructions are, they look more like that too. We believe that we can achieve better compression results with $p > 1024$, since this would allow us to better differentiate the classes. However, one major impediment to this approach is that the trade-off between the pure reconstruction and the utility for the specific downstream task must be included through a careful loss design.

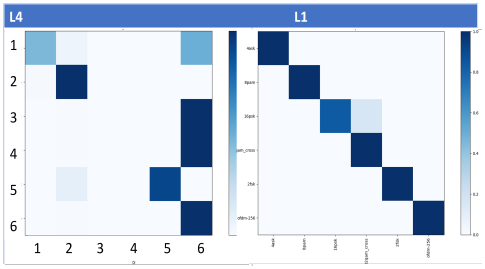

Fig. 10. Confusion matrices of the HQARF (one random training example) for L1 and L4 show that not all modulations are equally classifiable.

V. Conclusions and Future Work

In this paper we evaluate our model, HQARF, which is the first vector-quantization (VQ) based learned compression (LC) of modulated RF signals. The HQARF lossy reconstructions are evaluated through waveform plots, spectrograms and iq scatterplots. Finally, they are evaluated on a modulation recognition (ModRec) task, illustrating the utility of LC in this domain. We point out to the complex factors affecting the ModRec accuracy on the LC reconstructions, and the fidelity of their temporal plots, complex-plane scatterplots and spectrograms. These factors include the ability of loss functions to convey a physics-based model of the domain without knowing much about the data, the LC model architecture, training methodology, and the dimension and training of the VQ codebook. We suggest that a trade-off between the reconstruction quality and the utility should be mechanized through a more complex loss function, and that a longer datapoint be used. HQARF is a proof of concept architecture deserving further investigation, as it may have applications in intelligent spectrum management and may help leverage the *VQ based diffusion models* [11], [15] in this domain, possibly towards an AI-native air-interface. Finally, observe that we did not leverage the well-known denoising properties of autoencoders here: we are upgrading HQARF training in that direction, to be able to compress channel-distorted signals in addition to the high-SNR samples considered here.

References

[1] 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.

[2] 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.

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

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

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

[6] A. Rodriguez, Y. Kaasaragadda, and S. Kokalj-Filipovic, "Deep-Learned Compression for Radio-Frequency Signal Classification," 2024. [Online]. Available: https://arxiv.org/abs/2403.03150

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

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

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

[10] 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

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

[12] J. Austin et al., "Structured denoising diffusion models in discrete state-spaces," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[13] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, 2019.

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

[15] Yang, Ling et al., "Diffusion Models: A Comprehensive Survey of Methods and Applications," *ACM Computing Surveys*, vol. 56, no. 4, Nov 2023.