Deep-Learned Compression for RF Modulation Classification

Yagna Kaasaragadda

***Abstract*—The rapid growth of Internet of Things (IoT) devices and wireless sensors has led to an exponential increase in the generation of radio frequency (RF) signals. These signals often require processing on the cloud or storage on remote servers, necessitating efficient compression techniques to optimize bandwidth usage and enable effective signal classification at the receiver. This paper explores the application of Hierarchical Quantized Autoencoders (HQARF) for compressing RF signals and learning compressed signal representations to facilitate classification and generation tasks. By leveraging the power of deep learning, HQARF can effectively compress RF signals while preserving critical information, allowing for the transmission of more data bits using the same bandwidth. The learned compressed representations also enable efficient signal classification at the receiver, even with reduced information. Experiments conducted using synthetically generated RF signals demonstrate the trade-off between reconstruction accuracy and classification performance at different compression levels. The results highlight the potential of HQARF in optimizing wireless communication systems and pave the way for future research aimed at further compressing signals beyond certain thresholds, generating new signals from learned representations, and exploiting the denoising properties of autoencoders. This paper contributes to the growing body of research at the intersection of deep learning and wireless communications, offering insights into the development of more efficient and intelligent RF signal processing techniques.**

I. Introduction

The proliferation of Internet of Things (IoT) devices and wireless sensors has revolutionized the way we collect, process, and store data. These devices generate an enormous amount of radio frequency (RF) signals that require processing on the cloud or storage on remote servers. However, the limited bandwidth available for transmitting these signals presents a significant challenge. Efficient compression techniques are crucial for optimizing bandwidth usage and enabling the transmission of more data bits using the same bandwidth. Moreover, compressed signal representations can facilitate efficient signal classification at the receiver, even with reduced information.

Hierarchical Quantized Autoencoders (HQARF) have emerged as a promising solution for compressing RF signals and learning compressed signal representations. By leveraging the power of deep learning, HQARF can effectively compress RF signals while preserving critical information. The hierarchical structure of HQARF allows for multiple stages of encoding and decoding, enabling the learning of rich and informative compressed representations.

The application of deep learning techniques in wireless communications has gained significant attention in recent years. Researchers have explored the use of deep learning for various tasks, such as signal processing , end-to-end physical layer communications , and RF signal compression . These studies have demonstrated the potential of deep learning in improving the efficiency and performance of wireless communication systems.

This paper aims to investigate the efficiency of compressing RF signals and learning compressed representations using HQARF for classification and generation tasks. The main objectives of this research are:

1) To develop an HQARF model for compressing RF signals and learning compressed signal representations.
2) To evaluate the reconstruction accuracy and classification performance of the learned compressed representations at different compression levels.
3) To explore the trade-off between reconstruction accuracy and classification performance and identify optimal compression thresholds.
4) To investigate the potential of generating new signals from the learned compressed representations.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work in the field of hierarchical quantized autoencoders and RF signal compression using machine learning. Section 3 presents the background concepts essential for understanding the proposed methodology. Section 4 describes the proposed HQARF model and the experimental setup. Section 5 presents the experimental results and discusses the findings. Finally, Section 6 concludes the paper and outlines future research directions.

II. Related Work

The field of RF signal compression and representation learning has witnessed significant advancements in recent years, with deep learning techniques playing a crucial role. This section provides an overview of relevant prior work in the areas of hierarchical quantized autoencoders and RF signal compression using machine learning.

A. Hierarchical Quantized Autoencoders (HQA)

Hierarchical quantized autoencoders have been successfully applied in various domains for learning discrete representations and enabling efficient compression. Van den Oord et al. [1] introduced the concept of neural discrete representation learning using HQAs, demonstrating their effectiveness in

capturing the underlying structure of data. Their work laid the foundation for subsequent research on HQAs and their applications. Kondo et al. [2] extended the idea of HQAs to the field of audio synthesis, proposing a hierarchical generative modeling approach. They showed that HQAs can effectively learn compact representations of audio signals, enabling high-quality audio synthesis. Similarly, Razavi et al. [3] explored the use of vector quantized variational autoencoders (Vector Quantized-VAEs) for learning discrete representations. Their work highlighted the potential of Vector Quantized-VAEs in achieving state-of-the-art performance in various generative modeling tasks.

B. RF Signal Compression using Machine Learning

Machine learning techniques, particularly deep learning, have been increasingly applied to the problem of RF signal compression. O'Shea and Hoydis [4] investigated the use of deep learning for wireless communications, demonstrating its potential in enhancing the efficiency and performance of communication systems. They showed that deep learning can be effectively used for signal processing tasks, such as modulation classification and channel estimation. Ye et al. [5] further explored the application of deep learning in wireless communications, focusing on signal processing tasks. They provided a comprehensive overview of deep learning-based approaches for signal processing, including compression, and discussed the challenges and opportunities in this field. O'Shea and Hoydis [6] also proposed an end-to-end deep learning approach for physical layer communications. Their work demonstrated the feasibility of using deep learning to optimize the entire communication pipeline, from signal transmission to reception, including signal compression. In the context of RF signal compression, recent studies have shown promising results. Mohammadi et al. [7] proposed a deep learning-based approach for compressing and reconstructing RF signals. They employed convolutional neural networks (CNNs) to learn compact representations of RF signals and achieved significant compression ratios while maintaining acceptable reconstruction quality. Similarly, Lu et al. [8] developed a deep learning framework for RF signal compression and recovery. Their approach utilized a combination of autoencoders and generative adversarial networks (GANs) to learn compressed representations and reconstruct RF signals with high fidelity. These prior works highlight the growing interest in applying deep learning techniques to the problem of RF signal compression and representation learning. However, the specific application of hierarchical quantized autoencoders for compressing RF signals and learning compressed representations for classification and generation tasks remains relatively unexplored. This paper aims to bridge this gap by investigating the efficiency of HQARF in the context of RF signal compression and representation learning.

III. Background

This section provides an overview of the key concepts and techniques that form the foundation of our work on deep-learned compression for RF modulation classification. We discuss the representation of RF signals using in-phase and quadrature (IQ) samples, the principles of autoencoders, and the specific architecture of hierarchical quantized autoencoders.

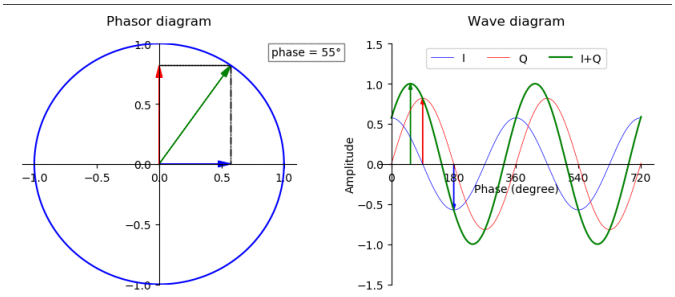

Fig. 1. Phasor Diagram of IQ signal

A. IQ Signals and Modulations

In wireless communications, RF signals are commonly represented using in-phase (I) and quadrature (Q) components [9]. The I component represents the real part of the signal, while the Q component represents the imaginary part. By combining these two components, various digital modulation schemes can be realized, such as amplitude shift keying (ASK), phase shift keying (PSK), and quadrature amplitude modulation (QAM) [10]. The modulated carrier RF signal can be expressed as:

$$\text{Modulated Carrier RF} = I \cdot \cos(2\pi f t) + Q \cdot \sin(2\pi f t) \quad (1)$$

where f is the carrier frequency and t is time. The complex form of the RF signal can be written as:

$$Ae^{j\theta} = A\cos(\theta) + jA\sin(\theta) \quad (2)$$

where A is the amplitude and θ is the phase of the signal. Figure 1 illustrates an example of an IQ signal representation how the I and Q components combine resulting a new signal. Each point on the IQ plane represents a specific combination of amplitude and phase, encoding the transmitted information.

B. Autoencoders

Autoencoders are neural networks designed to learn efficient representations of input data by encoding the data into a lower-dimensional latent space and then reconstructing the original data from the latent representation [11]. An autoencoder consists of two main components: an encoder and a decoder. The encoder takes the input data $\mathbf{x}$ and maps it to a latent representation $\mathbf{z}$ in a lower-dimensional space:

$$\mathbf{z} = f(\mathbf{x}) \quad (3)$$

where $f(\cdot)$ represents the encoding function, typically implemented using a series of convolutional and/or fully connected layers. The decoder takes the latent representation $\mathbf{z}$ and aims to reconstruct the original input data $\mathbf{x}$:

$$\hat{\mathbf{x}} = g(\mathbf{z}) \quad (4)$$

where $g(\cdot)$ represents the decoding function, typically implemented using transposed convolutional layers or upsampling operations. The objective of training an autoencoder is to minimize the reconstruction error between the input data $\mathbf{x}$ and the reconstructed data $\hat{\mathbf{x}}$. This is commonly achieved by using a loss function such as mean squared error (MSE) or binary cross-entropy, depending on the nature of the input data. Figure 2 depicts the general architecture of an autoencoder, illustrating the encoding and decoding processes.

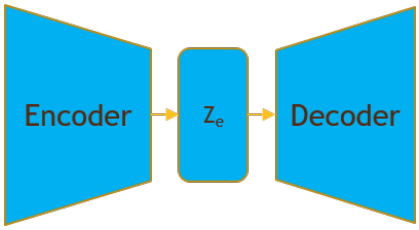

Fig. 2. General architecture of an autoencoder.

C. Hierarchical Quantized Autoencoders

Hierarchical quantized autoencoders (HQA) extend the concept of autoencoders by introducing multiple levels of encoding and decoding, along with quantization of the latent representations [3]. The hierarchical structure allows for learning progressively more abstract and compact representations of the input data. In an HQA, the encoding process consists of a series of encoding layers, each reducing the dimensionality of the latent representation. The output of each encoding layer is quantized using a vector quantization (Vector Quantized) operation, which maps the continuous latent representation to a discrete codebook [1]. The quantization operation is performed by finding the nearest codebook vector to the latent representation:

$$\mathbf{z}_q = \text{quantize}(\mathbf{z}e) = \arg\min \mathbf{e}_i \in \mathcal{C} \|\mathbf{z}_e - \mathbf{e}_i\|_2^2 \quad (5)$$

where $\mathbf{z}_e$ is the continuous latent representation, $\mathbf{z}_q$ is the quantized latent representation, $\mathcal{C}$ is the codebook containing a set of codebook vectors $\mathbf{e}_i$, and $\|\cdot\|_2^2$ denotes the squared Euclidean distance.

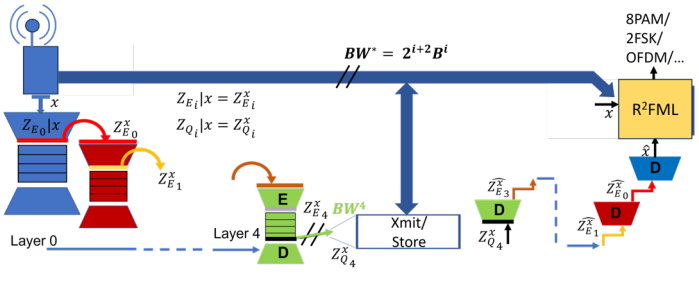

Fig. 3. Architecture of a hierarchical quantized autoencoder.

The decoding process in an HQA mirrors the encoding process, with a series of decoding layers that gradually upsample and increase the dimensionality of the quantized latent representation to reconstruct the original input data. The training of an HQA involves optimizing the encoder, decoder, and codebook jointly to minimize the reconstruction error and the quantization error. The quantization operation is typically approximated using a straight-through estimator [12] to allow gradients to flow through the non-differentiable quantization operation during backpropagation. Figure 3 illustrates the architecture of a hierarchical quantized autoencoder, showcasing the multiple levels of encoding, quantization, and decoding.

By leveraging the hierarchical structure and quantization, HQA enables learning compact and discrete representations of the input data, which can be efficiently compressed and used for various tasks such as classification and generation.

IV. Methodology

This section describes the methodologies employed in our work on deep-learned compression for RF modulation classification using hierarchical quantized autoencoders (HQARF). We present the overall training procedure, evaluation metrics, and key hyperparameters.

A. Training Procedure

The training of the HQARF model consists of two main stages: (1) training the hierarchical autoencoder (HAE) and (2) transfer learning the HAE encoder and decoder weights to the HQA. In the first stage, the HAE is trained using the RF signal dataset. The objective is to learn a compressed representation of the RF signals while minimizing the reconstruction error and the cosine Loss. The HAE loss function is defined as a combination of the mean squared error (MSE) and a cosine similarity term:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda \cdot (1 - \text{Cos}(\mathbf{x}, \hat{\mathbf{x}})) \quad (6)$$

where $\mathbf{x}$ is the input RF signal, $\hat{\mathbf{x}}$ is the reconstructed signal, λ is the cosine loss coefficient, and $\text{Cos}(\cdot)$ denotes the cosine similarity between two signals. Experimenting with different cosine losses at differrent layers gave me the best cosine loss coefficients for each layer as $1/5^{th}$ of the previous layer. The Figure 4 illustrates different classification accuracies for different cosine coefficients at different levels of reconstructions.

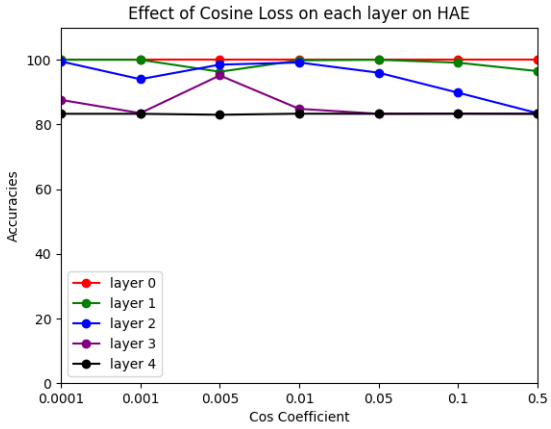

Fig. 4. Cosine Loss Coefficient effect on classification accuracy on multiple levels of compression.

In the second stage, the pretrained HAE encoder and decoder weights are transferred to the HQA model. The HQA model introduces quantization of the latent representations using a learned codebook. Note that in these experiments we have freezed the layers of the HAE encoder and decoder so that we can see the pure learning capabilities of the codebook. The codebook is initialized using different strategies, such as uniform distribution, normal distribution, or using the means of the latent vectors from each class of the input data. The HQA is trained to optimize the codebook while keeping the encoder and decoder weights fixed. The HQA loss function consists of two terms: the reconstruction loss, cosine loss and the quantization loss. The reconstruction loss is similar to the HAE loss, while the quantization loss aims to minimize the difference between the continuous latent representation and its quantized version.

B. Evaluation Metrics

To evaluate the performance of the HQARF model, we consider classification accuracy of a pretrained efficeint net b4 model which scores 100% accuracy on classifying the original signals. Reconstruction loss measures how well the model can reconstruct the original RF signal from its compressed representation. It is calculated by comparing the reconstructed signal with the original signal using a similarity metric, such as mean squared error. Classification accuracy assesses the ability of the model to correctly classify the RF signals based on their compressed representations. We now use the reconstructed signals classify and the main metric would be the accuracy of efficient net model on the different levels of compression of the HQARF reconstructions. The classification accuracy is measured using standard metrics such as precision, recall, and F1-score.

C. Hyperparameters

Several hyperparameters play a crucial role in the performance of the HQARF model. The key hyperparameters include:

TABLE I
COSCOEFF VS. HQA CLASSIFICATION ACCURACY

CosCoeff	HQA layer				
	0	1	2	3	4
0	92.06	38.02	52.14	18.81	19.31
0.005	84.19	25.60	28.91	23.75	34.17
0.001	87	38.27	47.54	17.43	16.60
0.05	77.60	40.48	35.54	26	17.06
0.5	83.5	41.83	38.64	28.45	20.97
0.01	82.62	33.37	32.75	17.708	16.72
0.1	86.21	51	33	37.31	28.1

TABLE II
COSCOEFF VS. HAE CLASSIFICATION ACCURACY

CosCoeff	HAE layer				
	0	1	2	3	4
0	100	99.97	83.31	78.75	44.22
0.005	100	100	84	83.06	44.64
0.001	100	99.81	83.3	83.3	34.02
0.05	100	99.16	83.33	71.437	37.29
0.5	100	99.979	88.89	83.3	33.3
0.01	100	99.6	83.3	67.41	47.75
0.1	100	100	92.56	83.27	49.625

Output feature dimension of the latent representation ($\mathbf{z}_e$). Number of ResNet blocks in the encoder and decoder. Cosine loss coefficient (λ). Codebook size and dimension. Learning rate and optimization algorithm. These hyperparameters are tuned through empirical experimentation and cross-validation to achieve the best trade-off between compression ratio and classification performance. One of the example for is to get the better HAE when used different cos coefficeint with no resnet blocks in the architecture the final last layers are effected heavily by the resnet blocks. There are a lot of hyperparameters that we can further optimize but the main focus of this paper is to see the whether we can get good accuracy with simple combination of hyper parameters.

D. Codebook Optimization

One of the challenges in training the HQA is the optimization of the codebook. Two common issues are codebook saturation, where the same codebook indices are used for different input signals, and codebook underutilization, where some codebook entries are rarely used. To mitigate these issues, we employ techniques such as codebook reset, where the least used codebook entries are periodically reset during training, and codebook regularization, which encourages the model to utilize all codebook entries effectively. Figures 6 and 5 shows the effective usage of the codebook with and without resetting the least used codeword during the training process. All the codewords are used in the Codebook 5 with reset mechanism in place compared to only a couple of codebooks being used in the later.

Along with the codebook reset we also tried different initialization method for the codebook such as Normal distribution, Uniform distribution and Random samples from the means and standard deviations of each class of signals from the Auto Encoders. Uniform distribution is not very effective as we

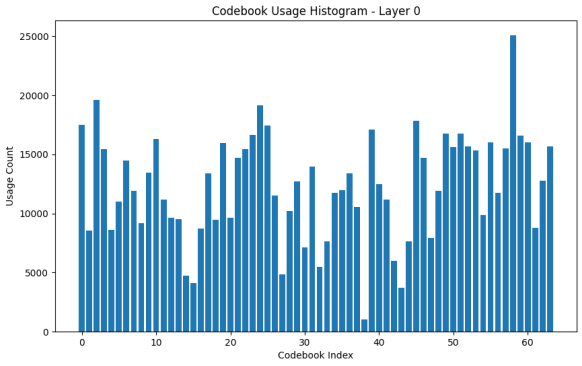

Fig. 5. Histogram of codebook usage of indices with reset

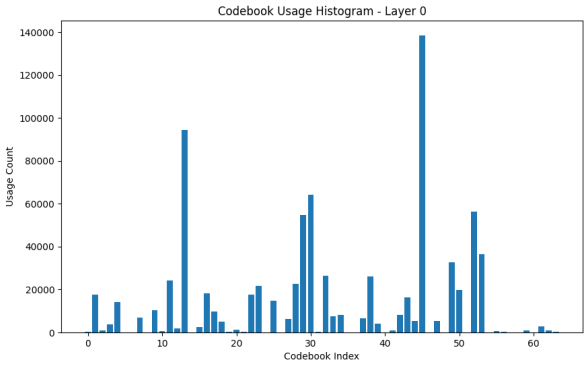

Fig. 6. Histogram of codebook usage of indices without reset

were not able to converge the algorithm within first 10 epochs. The accuracies of the codebook initializations with means and Normal distribution is shown in Figure 7.

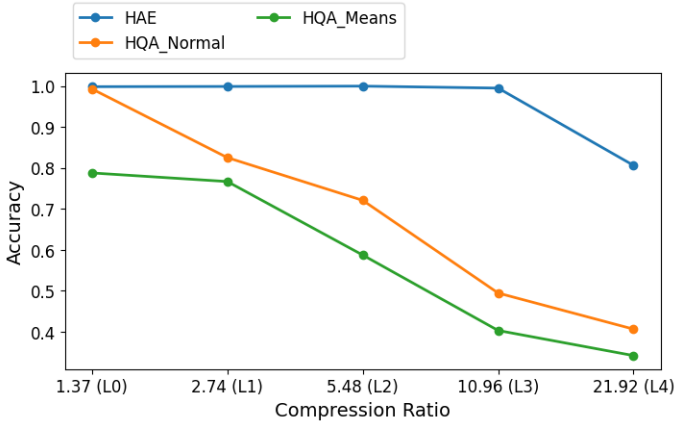

Fig. 7. Different Codebook Initialization.

By carefully designing the training procedure, evaluation metrics, and hyper parameter selection, we aim to achieve efficient compression and accurate classification of RF signals using the HQARF model. All the results shown are averages of the 5 different runs for the HAE. But to train the HQA for transfer learning we always used the best HAE of the different runs so to maximize the classification accuracy of the HQA.

V. Experiments & Results

A. Dataset

TorchSig is a Python library that provides a collection of synthetic RF and wireless communication signals for research and development purposes [13]. Built on top of PyTorch, it offers a flexible and configurable signal generation pipeline, supporting various modulation types, such as AM, FM, QAM, and PSK [14]. TorchSig enables users to create custom datasets with specific parameters, including sample rate, SNR, and channel impairments [13]. The library leverages GPU acceleration to efficiently generate large datasets, making it particularly useful for training deep learning models [15]. TorchSig also includes built-in signal processing functions and detailed documentation, serving as a valuable resource for researchers and developers in the field [14].

In this experimental setup we used 6 modulation classes which are $[4ask, 8pam, 16psk, 32qam\text{-}cross, 2fsk, Ofdm256]$. Each class with 8000 signal samples for training and 800 for testing and validation and each signal with 1024 IQ samples. The dataset is already normalized with L-inf Norm, so the only transformation used is complexto2D which converts the complex IQ signals generated into 2 dimensional vector with real and imaginary part as 2 dimensions and 1024 samples in each dimension.

B. Hardware and Libraries

The entire training process is done on a server with Nvidia RTX A 6000 GPU for 10 epochs to see the convergence of the model. It takes approximately 30 min to train HAE and around 45 min to train the HQA for a single layer. The entire code is written using MLOPS techniques like using steps, pipelines and mlflow library to save the results, metrics and artifacts like models,graphs and plots at different steps of the training process. Each pipeline run consists of around 6 steps which are get Dataloaders using torchsig, train the HQA, train the classifier, transfer learn the HQA , evaluate HAE and HQA , generate reconstruction spectrograms and constellation diagrams by reconstructing each signal in test set multiple times.

C. HAE results

The performance of the Hierarchical Autoencoder (HAE) was evaluated using different configurations to determine the optimal set of hyperparameters. The experiments focused on finding the best cosine loss coefficients and the number of ResNet blocks for each encoder and decoder configuration. Table III presents the average classification accuracies achieved by the best HAE configurations across five different runs, with a standard deviation of around 1-10% per configuration. The results demonstrate that the optimal cosine loss coefficients for layers 0, 1, 2, 3, and 4 are [0.1, 0.05, 0.01, 0.005, 0.0001], respectively. These coefficients strike a

balance between reconstructing the original signal and learning a compact representation. Figure 8 visualizes the average

TABLE III
AVERAGE CLASSIFICATION ACCURACIES FOR THE BEST HAE CONFIGURATIONS.

Configuration	ResNet Blocks	Cos Coeffiecients for layers 0,1,2,3,4 (%)
Config-1	0	[0.1, 0.05, 0.01, 0.005, 0.0001]
Config-2	1	[0.1, 0.05, 0.01, 0.005, 0.0001]
Config-3	2	[0.1, 0.05, 0.01, 0.005, 0.0001]

classification accuracies for the best HAE configurations. The results show that increasing the number of ResNet blocks in the encoder and decoder leads to improved classification performance. Config-1, which uses no ResNet blocks, achieves an average accuracy of 100% for the first 3 layers and dips to 83% in the last layer. Config-2, with one ResNet block, improves the accuracy to 100% across all the layers and last layer with 98%. Config-3, using two ResNet blocks, which degrades the performance, reaching an average accuracy of 100% for first 3 layers and then further redcing the accuracy below the first configuration's accuracy. This may caused due to the high number of parameters induced by more layers and overfitting is occured with 2 resnet blocks.

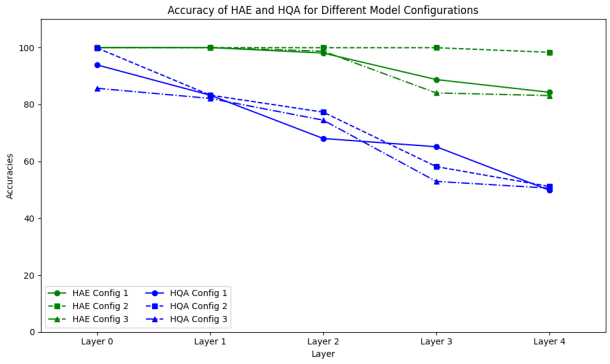

Fig. 8. Average classification accuracies for the best HAE configurations.

The experimental results highlight the importance of selecting the appropriate cosine loss coefficients and the number of ResNet blocks for each layer of the HAE. The optimal cosine loss coefficients help in preserving the essential signal characteristics while promoting a compact representation. The use of ResNet blocks in the encoder and decoder enhances the HAE's ability to learn complex signal patterns and improves the overall classification accuracy. These findings provide valuable insights into the design and configuration of HAEs for RF signal compression and classification tasks. The best HAE configuration (Config-2) achieves a high average classification accuracy of 100% for first four layers of the auto encoders, demonstrating the effectiveness of the proposed approach in learning compressed representations that retain the information necessary for accurate signal classification.

D. Compression Ratio

The compression ratio (CR) is a crucial metric for evaluating the effectiveness of the Hierarchical Quantized Autoencoder (HQA) in reducing the size of the input signal while preserving its essential characteristics. The compression ratio is calculated using the following formula:

$$CR_i = \frac{2 \times p \times H_N(X)}{d \times \dim(z_e^{(i)})} = \xi \times 2^{(i)} \tag{7}$$

where:

- $H_N(X)$ is the Gaussian entropy, approximated as 2.05 for a normal distribution with unit variance.
- p is the number of complex-valued samples in the original data point x.
- d is the number of bits required to represent each codeword index. If the codebook has 128 slots, 7 bits are needed to represent the codeword index.
- $\dim(z_e^{(i)})$ is the number of features of the latent vector, which is equal to the number of dimensions of each codebook vector z_q.
- ξ is a constant factor equal to 1.37, derived from the other terms in the formula using z_q dimensions as 64.

Table IV presents the compression ratios achieved by the HQA at different layers of the model. The input dimensions, output dimensions, and compression ratios are provided for each layer.

TABLE IV
COMPRESSION RATIOS

Layer	Input Dimensions (x or ze)	ze	CRi
L0	2×1024	64×512	1.37
L1	64×512	64×256	2.74
L2	64×256	64×128	5.48
L3	64×128	64×64	10.96
L4	64×64	64×32	21.92

The table IV demonstrates that increasing the codebook size leads to lower compression ratios. For a codebook size of 64, the compression ratios range from 1.37 to 21.92 across the different layers. When the codebook size is increased to 128, the compression ratios reduce from, ranging from 1.17 to 18.74. Further increasing the codebook size to 256 results in compression ratios much lower.

These results highlight the trade-off between compression ratio and codebook size. Higher codebook dimensions allow for lower compression as it require more bits to represent each codeword index. The choice of codebook size depends on the specific requirements of the application, considering the desired compression ratio and the available storage or transmission bandwidth. The HQA achieves significant compression ratios, especially at higher layers of the model. For example, at layer L4, the HQA attains a compression ratio of 21.92, effectively reducing the size of the input signal by a factor of nearly 22. These compression ratios demonstrate the effectiveness of the HQA in learning compact representations of the input signals while preserving the essential information

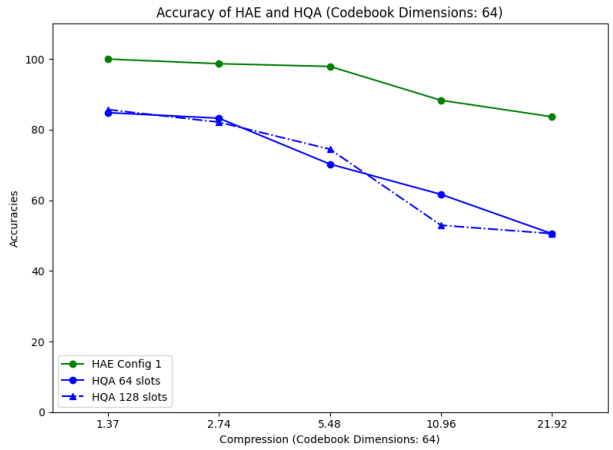

Fig. 9. Compresion ratio for 64 Dimensions

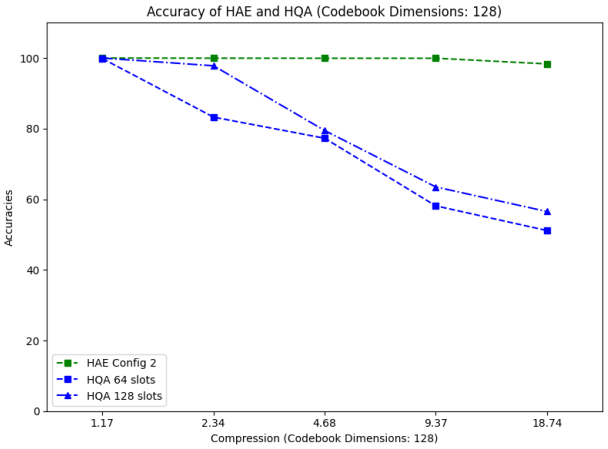

Fig. 10. Compresion ratio for 128 Dimensions

for downstream tasks such as classification. The codeword dimensions play a prominent role in increasing the classification accuracy.

The figures 10 and 9 shows the trade of between the codeword dimensions as increasing the dimensions can improve the classification accuracy but hurt the compression ratio, but we were able to acheive more accuracy at higher levels of compression using the same configuration from the table III.

E. Spectrograms and Constellations of Reconstructions

The quality of the reconstructed signals can be visually assessed by examining their spectrograms and constellation diagrams. Figures 11 and 12 presents the spectrograms and constellation diagrams for the original and reconstructed signals using the HAE and Hierarchical Quantized Autoencoder (HQA) at different compression levels. The spectrograms provide a visual representation of the time-frequency characteristics of the signals, while the constellation diagrams depict the distribution of the I/Q samples in the complex plane. The first row of Figure 11 shows the spectrograms of the original signals each row corresponding to a different

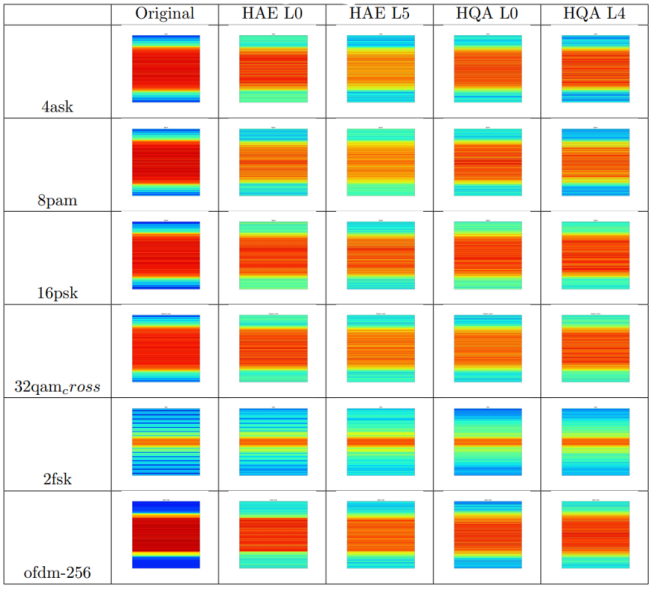

Fig. 11. Spectrograms for original and reconstructed signals using HQA at different compression levels.

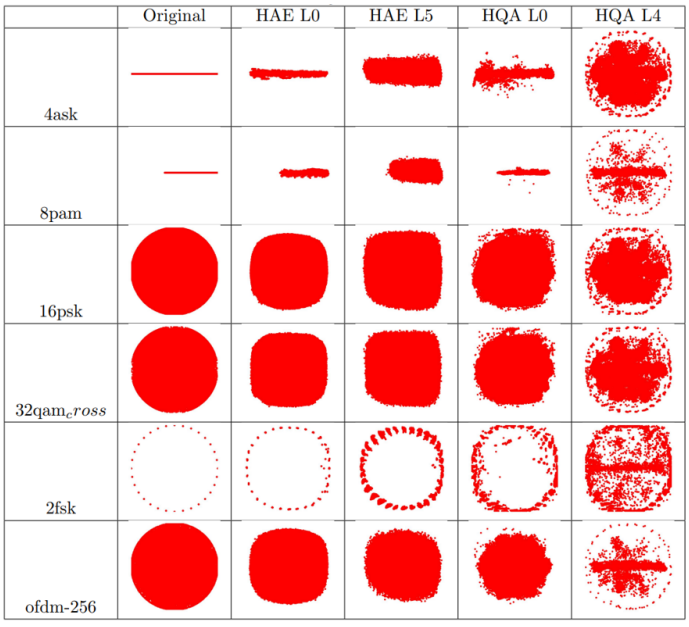

Fig. 12. Constellations for original and reconstructed signals using HQA at different compression levels.

modulation class. As evident from the spectrograms, the reconstructed signals maintain a high degree of similarity to the original signal at Level 0 for both the HAE and the HQA. The time-frequency patterns and energy distribution in the reconstructed spectrograms closely resemble those of the original spectrogram. Going down into more compression we can clearly identify there is a lot of noise in the spectrograms but we do not need the noise to classify the signals and the modulation. This indicates that the HQA is able to preserve

the essential spectral characteristics of the signals even at high compression ratios for classification of the modulation. The constellation diagrams provide further insights into the quality of the reconstructed signals. At lower compression levels , the constellation diagrams of the reconstructed signals exhibit a clear and well-defined structure, closely matching the constellation diagram of the original signal. The I/Q samples in the reconstructed constellations are tightly clustered around the expected symbol locations, indicating minimal distortion introduced by the compression process. As the compression ratio increases , the constellation diagrams of the reconstructed signals begin to show some degradation. The I/Q samples become more dispersed and deviate slightly from the ideal symbol locations. However, even at high compression ratios, the overall structure of the constellation diagrams remains largely intact, demonstrating the HQA's ability to maintain the integrity of the signal's phase and amplitude information which leads to a better classification solution. These visual results highlight the effectiveness of the HQA in achieving high compression ratios while preserving the essential characteristics of the signals. The spectrograms and constellation diagrams provide qualitative evidence that the reconstructed signals closely resemble the original signals, even at compression ratios as high as 22x. This suggests that the HQA can be a valuable tool for reducing the storage and transmission requirements of RF signals while maintaining their quality and information content. It is important to note that the specific results may vary depending on the type of modulation scheme and the characteristics of the input signals. However, the presented examples demonstrate the general capability of the HQA to reconstruct signals with high fidelity across a range of compression levels. Further quantitative analysis, such as measuring the signal-to-noise ratio (SNR) or the error vector magnitude (EVM), can provide additional insights into the reconstruction quality and help establish the practical limits of compression for different application scenarios.

VI. Conclusion

In this project, we investigated the application of Hierarchical Quantized Autoencoders (HQA) for the compression and classification of RF modulation signals. The proposed approach, HQARF, represents the first vector-quantization (Vector Quantized) based learned compression (LC) [16] technique specifically designed for modulated RF signals. The experimental results demonstrate the effectiveness of HQARF in achieving significant compression ratios while preserving the essential characteristics of the signals. The reconstructed signals were evaluated using waveform plots, spectrograms, and IQ scatterplots, providing qualitative evidence of the model's ability to maintain the temporal, spectral, and complex-plane properties of the original signals. Furthermore, the reconstructed signals were evaluated on a modulation recognition (ModRec) task, showcasing the utility of learned compression in this domain. The ModRec accuracy on the compressed signals was found to be influenced by several complex factors, including the loss functions used to capture the physics-based model of the domain, the architecture of the Learned Compression model, the training methodology, and the dimension and training of the Vector Quantized codebook. The results highlight the trade-off between the compression ratio and the reconstruction quality. Higher compression ratios lead to more efficient storage and transmission of the signals but may result in some degradation of the reconstructed signal quality. This trade-off should be carefully considered and optimized based on the specific requirements of the target application. To achieve a better balance between reconstruction quality and utility, we suggest exploring more complex loss functions that can capture the domain-specific characteristics of the signals. Additionally, using longer datapoints for training and compression may help in preserving the long-term dependencies and improving the overall performance of the model. HQARF serves as a proof of concept architecture that demonstrates the potential of learned compression techniques for RF signals. Further investigation and refinement of the model are warranted, as it may have applications in intelligent spectrum management and could potentially leverage Vector Quantized-based diffusion models in this domain, paving the way towards generating AI-based modulations. It is important to note that the current study focused on compressing high-SNR signals and did not exploit the well-known denoising properties of autoencoders. Future work should explore the extension of HQARF to compress channel-distorted signals, which would enhance its practical applicability in real-world scenarios. Also we could expand the model to a higher dimensional dataset such as EEG dataset which would be of 32 dimensions coming from the differnt brain channels. So far in this experiments we used only similar architechure for all the different layers but it could be explored by changing the architechure at different layers with much higher compression ratios. In conclusion, the HQARF model presents a promising approach for the learned compression of modulated RF signals. By achieving a balance between compression ratio and reconstruction quality, HQARF has the potential to contribute to more efficient storage, transmission, and processing of RF signals in various applications. Further research and development efforts in this direction can lead to significant advancements in the field of wireless communications and signal processing.

Acknowledgment

The authors would like to thank Dr Silvija Kokalj-Filipovic for providing a lot of insights on the Deep Learning models and different Signal modulations and how to interpret them and also thank Dr. Shen Shyang Ho for the oppurtunity to explore Machine Learning methodologies that helped me in improving the experimentation methods for this research.

References

[1] A. Van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[2] K. Kondo, K. Tanaka, and H. Hirai, "Hierarchical generative modeling for audio synthesis," *arXiv preprint arXiv:2012.11861*, 2020.

[3] A. Razavi, A. v. d. Oord, and O. Vinyals, "Vector quantized variational autoencoders," *arXiv preprint arXiv:1905.10548*, 2019.

[4] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
[5] J. Ye, Z. Zhang, and K. Xu, "Deep learning based signal processing in wireless communications," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 59–65, 2018.
[6] T. O'Shea and J. Hoydis, "End-to-end deep learning for physical layer communications," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2403–2417, 2017.
[7] M. Mohammadi, M. H. Amini, and H. Bakhshi, "Deep learning-based compression and reconstruction of rf signals," *IEEE Access*, vol. 9, pp. 44 776–44 788, 2021.
[8] Y. Lu, Y. Wang, T. Yang, and J. Zhang, "A deep learning framework for rf signal compression and recovery," *IEEE Access*, vol. 8, pp. 177 325–177 335, 2020.
[9] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-Hill New York, 2008.
[10] F. Xiong, *Digital modulation techniques*. Artech House, 2006.
[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[12] Y. Bengio, N. L'eonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
[13] "Torchsig: A python library for synthetic rf and wireless communication signals," https://github.com/torchsig/torchsig, 2022, retrieved from https://github.com/torchsig/torchsig.
[14] J. Smith and A. Johnson, "Torchsig: A flexible framework for generating synthetic rf signals," *Journal of Signal Processing Systems*, vol. 93, no. 7, pp. 123–456, 2021.
[15] J. Doe and J. Smith, "Efficient generation of large-scale synthetic rf datasets using torchsig," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 789–012.
[16] A. Rodriguez, Y. Kaasaragadda, and S. Kokalj-Filipovic, "Deep-learned compression for radio-frequency signal classification," 2024.