Automatic Modulation Classification using Quantum-Empowered Training

To Truong An[1][0000−0002−9394−3387], Simon L. Cotton[1][0000−0003−2620−6501], Hans-Jürgen Zepernick[2][0000−0003−3604−2766], and Trung Q. Duong[1,3][0000−0002−4703−4836]

[1] Centre for Wireless Innovation (CWI), Queen's University Belfast, Belfast, BT3 9DT, U.K.
[2] Blekinge Institute of Technology, 37179 Karlskrona, Sweden.
[3] Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada.
{ato01, simon.cotton, trung.q.duong}@qub.ac.uk, hans-jurgen.zepernick@bth.se.

Abstract. Automatic modulation classification (AMC) is a technique designed to identify the modulation of unknown wireless signals. However, existing AMC systems are resource-intensive since they rely on deep learning (DL) models. Training an AMC model is particularly demanding because a large number of parameters need to be updated repeatedly. In this work, we introduce quantum-empowered training (QET), a framework that enhances the training efficiency of AMC systems. The QET framework is a hybrid quantum classical neural network which aims to generate the classical parameters for the AMC model during training. This indirect training strategy significantly reduces the number of trainable parameters and the overall memory requirements compared to traditional training methods for DL models. The experimental results demonstrate that the QET enables training an AMC model with 70% fewer trainable parameters than classical training methods, while maintaining a competitive classification accuracy.

Keywords: Automatic modulation classification · Quantum machine learning · Training efficiency.

1 Introduction

Automatic modulation classification (AMC) is a technique used to identify the modulation scheme of received signals in non-cooperative communication environments, which is widely applied in both military and civilian domains [1, 2]. In military settings, AMC supports signal interception and analysis, while in civilian systems, it improves transmission reliability and data throughput by adapting modulation schemes to changing channel conditions [3]. To accurately classify the modulations of transmitted signals in complex wireless environments, deep learning (DL) has emerged as the dominant approach for AMC. Models

such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have demonstrated strong capabilities in extracting discriminative features and achieving high classification accuracy [4–7]. O'Shea et al. introduced the RML2016.10a dataset and proposed a CNN-based model for effective modulation classification [8]. Since then, numerous studies have investigated the robustness of AMC models, focusing on their classification accuracy, resilience to channel variations, and inference efficiency [9, 10].

With the advent of the sixth generation (6G) of mobile networks, the number of connected devices and services is expected to grow dramatically, placing greater demand on base station (BS) resources [11]. AMC training is typically executed on a BS server, which may only be allocated a small fraction of memory, as the server also supports various other network functions and applications. However, training an AMC model is inherently resource-intensive, especially in terms of memory utilization. The total memory footprint required for training includes not only memory for storing the model parameters, but also the associated intermediate activations and gradients that need to be stored for backpropagation [12, 13]. Consequently, the traditional training approach is inefficient and is often impractical in a resource-constrained environment. Although techniques such as model pruning aim to reduce model size during inference without compromising accuracy, they fail to address the main bottleneck caused by *training inefficiency* [14, 15]. This highlights a critical gap, namely a lack of solutions that are not only accurate but also, computationally efficient during training.

In this context, quantum machine learning (QML) shows much promise. Building on quantum principles such as superposition, entanglement, and parallelism, QML provides an effective approach for handling high-dimensional and noisy signals [16–18]. Experimental and theoretical studies suggest that QML can improve accuracy, lower model complexity, and accelerate convergence in specific applications [19, 20]. However, QML is still in its early stages and faces many challenges, especially in encoding large-scale datasets, where qubit coherence limits and circuit complexity pose significant barriers [21, 22]. Practical deployment is also limited by the reliance on quantum hardware for inference, which remains impractical due to the scarcity of scalable quantum computers.

One promising direction to overcome these challenges is the QuantumTrain (QT) framework, first introduced for image classification tasks [23]. QT uses a quantum parameter generator (QPG), which combines a quantum neural network (QNN) with a classical mapping network to produce trainable parameters for DL models. Unlike many QML approaches, QT processes inputs entirely in the classical domain, thereby circumventing the costly step of encoding large datasets into quantum states. The trained model itself remains classical, meaning inference can be performed on standard hardware without requiring access to quantum computers. At the same time, the use of QML during training improves efficiency and greatly reduces the number of parameters compared with traditional methods.

In this paper, we propose a novel training framework for AMC systems, called quantum-empowered training (QET), which utilizes a QNN and a mapping net-

work to generate weights and biases for a DL-based AMC during training. Instead of directly optimizing the full set of parameters in the target AMC model, only the QNN and mapping network are trained to produce these parameters. This indirect optimization paradigm reduces the number of trainable parameters and enhances training efficiency compared to conventional methods. QET builds on the QT framework. To adapt QT to AMC, we introduce a Multimodel mapping network that efficiently generates multiple classical parameters from a single QNN output. This design reduces qubit requirements and helps mitigate the barren plateau (BP) problem.

The main contributions of this work are summarized below:

- We propose QET, a framework designed to enhance the training efficiency in AMC systems. To the best of our knowledge, this is the first paper to directly address training inefficiency in AMC.
- We introduce a Multimodel mapping network that generates multiple classical parameters from a shared QNN output. By tuning the number of parameters mapped per output, the proposed mapping network reduces qubit requirements, employs multimodal learning for probability estimation, and helps mitigate the BP problem.
- We evaluated the performance of the proposed approach by utilizing QET to train an AMC model. Through our experiments, we demonstrated that QET can train an AMC model requiring only 30% of the original number of parameters, while still achieving comparable classification accuracy.

The structure of the paper is as follows. Section 2 describes the design of the QET framework, while Section 3 outlines the experimental setup. Section 4 presents and analyzes the results. Finally, Section 5 concludes the study.

2 Quantum-Empowered Training Framework

2.1 Overall Quantum-Empowered Training Framework

QET is a hybrid quantum-classical machine learning method designed to enhance training efficiency in AMC systems. As illustrated in Fig. 1, QET uses a QPG to produce parameters for an AMC model. The QPG contains three modules: a QNN; an embedding module; and a mapping network.

Fig. 1 presents a block diagram of the QNN architecture, illustrating the arrangement of the U_3 and controlled-U_3 (denoted CU_3) gates across Q qubits. The U_3 gate is used to manipulate quantum states and generate superposition. The U_3 gate may be represented as

$$U_3(\theta, \varphi, \lambda) = \begin{bmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\varphi}\sin(\theta/2) & e^{i(\varphi+\lambda)}\cos(\theta/2) \end{bmatrix}, \tag{1}$$

where θ, φ, λ are real parameters that define the rotation.

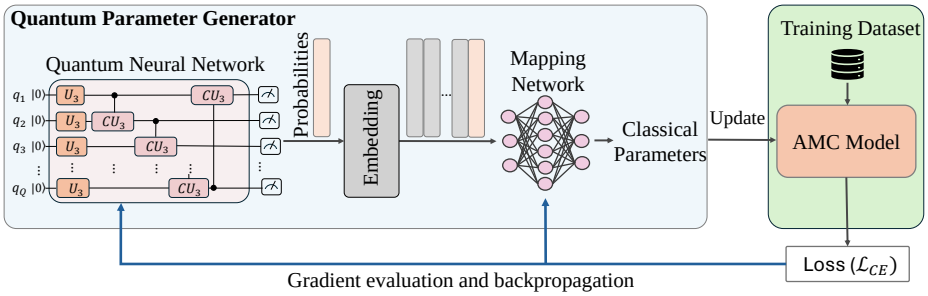

Fig. 1. The proposed quantum-empowered training framework.

To introduce entanglement among qubits, the QNN employs the CU_3 gate. This gate applies the U_3 operation to a target qubit, conditioned on the state of a control qubit. Its matrix form is expressed as follows [24].

$$CU_3(\tilde{\theta},\tilde{\varphi},\tilde{\lambda})_{q_c,q_t} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\frac{\tilde{\theta}}{2}\right) & 0 & -e^{i\tilde{\lambda}}\sin\left(\frac{\tilde{\theta}}{2}\right) \\ 0 & 0 & 1 & 0 \\ 0 & e^{i\tilde{\varphi}}\sin\left(\frac{\theta}{2}\right) & 0 & e^{i(\tilde{\varphi}+\tilde{\lambda})}\cos\left(\frac{\tilde{\theta}}{2}\right) \end{bmatrix}, \tag{2}$$

where q_c, q_t denote the control and target qubits, respectively.

The quantum state of the QNN can be expressed as

$$|\psi(\boldsymbol{\beta})\rangle = \left(\prod_r CU_3^{r,r+1}(\tilde{\theta},\tilde{\varphi},\tilde{\lambda})\prod_s U_3^s(\theta,\varphi,\lambda)\right)|0\rangle^{\otimes Q}, \tag{3}$$

where $\beta = \{(\tilde{\theta},\tilde{\varphi},\tilde{\lambda}),(\theta,\varphi,\lambda)\}$ denotes the trainable parameters of QNN, $|0\rangle^{\otimes Q}$ denotes the initial state of Q qubits, U_3^s are parameterized single-qubit rotations acting on qubit s, and $CU_3^{r,r+1}$ are controlled-U_3 gates that entangle neighboring qubits.

The measurement statistics of the QNN can be represented as a probability distribution over the computational basis. For a register of Q qubits, there are $\mathcal{P} = 2^Q$ possible outcomes. These probabilities can be arranged into the vector, Ψ,

$$\Psi = \begin{bmatrix} |\langle\phi_1|\psi(\boldsymbol{\beta})\rangle|^2 \\ \vdots \\ |\langle\phi_k|\psi(\boldsymbol{\beta})\rangle|^2 \\ \vdots \\ |\langle\phi_{\mathcal{P}}|\psi(\boldsymbol{\beta})\rangle|^2 \end{bmatrix}, \tag{4}$$

where $|\phi_k\rangle$ denotes the k-th computational basis state.

In the embedding stage, we form an augmented probability matrix by concatenating the measurement probabilities with the corresponding computational

basis states. Because downstream classical parameters can be both positive and negative values, each qubit state is assigned to -1 for $|0\rangle$ and $+1$ for $|1\rangle$. Consequently, the embedding matrix satisfies $\mathcal{E}(\Psi) \in [-1, 1]^{\mathcal{P}\times(Q+1)}$. The embedding matrix, $\mathcal{E}(\Psi)$, can be expressed as

$$\mathcal{E}(\Psi) = \begin{bmatrix} |\phi_1\rangle \parallel |\langle\phi_1|\psi(\beta)\rangle|^2 \\ \vdots \quad \vdots \qquad \vdots \\ |\phi_k\rangle \parallel |\langle\phi_k|\psi(\beta)\rangle|^2 \\ \vdots \quad \vdots \qquad \vdots \\ |\phi_{\mathcal{P}}\rangle \parallel |\langle\phi_{\mathcal{P}}|\psi(\beta)\rangle|^2 \end{bmatrix}, \tag{5}$$

where the $\parallel$ represents the concatenation operation. As an example, the first row of $\mathcal{E}(\Psi)$ is

$$\mathcal{E}(\Psi)_1 = |\phi_1\rangle \parallel \big|\langle\phi_1 \mid \psi(\beta)\rangle\big|^2 \tag{6}$$

$$= \big[\underbrace{-1, -1, \ldots, -1}_{Q \text{ entries}}, \big|\langle\phi_1 \mid \psi(\beta)\rangle\big|^2\big]. \tag{7}$$

Finally, the mapping network is a classical neural network that translates the augmented probability features into the weights and biases of the AMC model. Consider a classical AMC model with $\mathcal{C}_{\text{AMC}}$ trainable parameters, collected as $\theta_{\text{AMC}} = (\theta_1, \theta_2, \ldots, \theta_{\mathcal{C}_{\text{AMC}}})$. The mapping function is denoted G_{Θ}. The parameter vector generated by the QET is given by

$$\theta = G_{\Theta}\big(\mathcal{E}(\Psi)\big), \tag{8}$$

where θ is the QPG-produced parameter vector of length $\mathcal{C}_{\text{AMC}}$,Θ is the trainable parameters of the mapping network.

2.2 Proposed Mapping Network

Existing Mapping Network: In the QT framework proposed in [23, 24], the authors utilize a multilayer perceptron (MLP) as a mapping network to convert each QNN output directly to a corresponding parameter of the target model. The architecture of the MLP mapping network is shown in Fig. 2(a). The current mapping network relies on a one-to-one strategy, which is inefficient since training large DL models require many qubits. The number of qubits can be defined as

$$Q = \lceil \log_2 \ \mathcal{C}_{\text{AMC}} \rceil \, . \tag{9}$$

As the number of qubits increases, QNNs suffer from the BP problem, where the gradient of the cost function vanishes exponentially, making training extremely difficult [25].

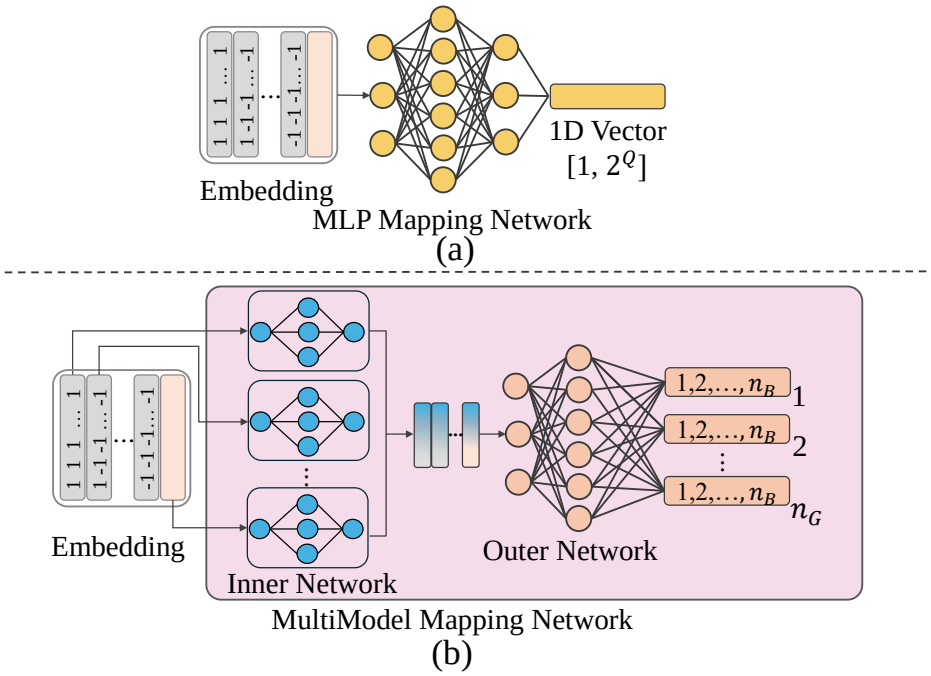

Fig. 2. Overview of the mapping network in the quantum-empowered training framework

(a) Existing architecture. (b) Proposed architecture.

Proposed Mapping Network: We use a multimodel mapping network to overcome these limitations and more effectively learn QNN measurement probabilities. We aim to enhance the mapping network by enabling each measurement probability to produce multiple classical parameters, instead of limiting it to a single parameter. This grouping strategy improves the efficiency of quantum information usage and decreases the total number of required qubits. Let n_B represent the number of classical parameters extracted from a single quantum representation. The number of groups is then given by $n_G = \lceil \mathcal{C}_{\text{AMC}}/n_B \rceil$, with each group containing n_B parameters.

As shown in Fig. 2(b), Multimodel mapping network consists of two main components. The first component is an inner network that processes each input variable independently, enabling the model to better capture QNN probability distributions by decoupling variable influences. The second component, an outer network, implemented as an MLP, takes the outputs of the inner network and forms a $[n_G, n_B]$ matrix. This matrix is subsequently flattened into a one-dimensional vector of length $n_G \times n_B$. The configuration of the mapping network as reported in Table 1.To center the distribution, the mean of the vector is removed from all elements, producing a zero-mean vector [24]. The resulting

Table 1. Detailed architecture of the *Multimodel mapping network* model

Module	Layer Description	Output Size	Parameters
Inner Network†	Linear	16	32
	Linear+ BatchNorm + ReLU	16	304
	Linear	1	17
Outer Network	Linear+ BatchNorm + ReLU	32	$32 \cdot Q + 128$
	Linear	16	528
Final Layers	Linear	16	272
	Linear	n_B	$17 \cdot n_B$

† *The inner function is applied independently to each input variables.*

centered vector constitutes the raw parameter values input to the classical AMC model.

Restructuring the parameter mapping process reduces the qubit requirement from $Q = \lceil \log_2 \mathcal{C}_{\text{AMC}} \rceil$ to $Q = \lceil \log_2 n_G \rceil$. Moreover, this design provides active control of the qubit count by tuning n_B. Within the QET framework, such control is particularly important, as it lowers qubit demand and thereby mitigates the BP problem.

2.3 Training Flow for the QET Framework

During each training iteration, the QPG outputs a full parameter set θ that substitutes the parameters of the AMC model. The updated AMC then processes a training input to produce predictions, which are evaluated using the cross-entropy (CE) loss. The CE loss quantifies the discrepancy between the predicted class probabilities $\hat{y}$ and the ground-truth labels y, and is defined as

$$\mathcal{L}_{CE} = -\frac{1}{N_{\text{Batch}}} \sum_{i=1}^{N_{\text{Batch}}} \sum_{c=1}^{N_c} y_{i,c} \log(\hat{y}_{i,c}), \tag{10}$$

where N_{Batch} is the batch size and N_c the number of classes.

Since the AMC parameters are produced by the QNN and the mapping network, their dependency can be expressed as $\theta = \theta(\beta, \Theta)$. In the training phase, both the QNN and mapping network parameters are optimized to minimize the CE loss. By employing the chain rule together with the parameter-shift rule, the gradient of the loss with respect to the combined parameters is obtained as [23]

$$\nabla_{\beta,\Theta} \mathcal{L}_{\text{CE}} = \left(\frac{\partial \theta}{\partial(\beta, \Theta)} \right)^{\top} \nabla_{\theta} \mathcal{L}_{\text{CE}}, \tag{11}$$

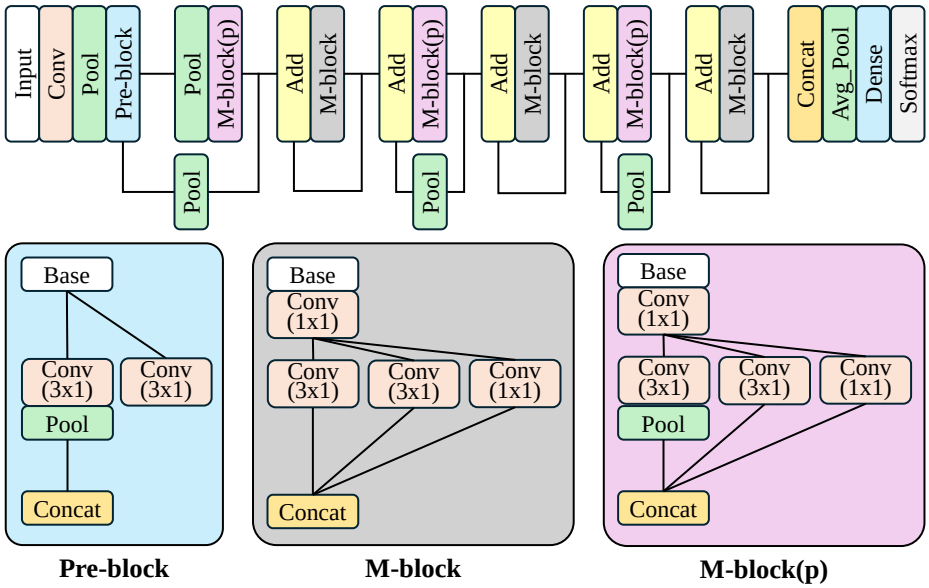

Fig. 3. Architecture of the MC-Net model [6].

where $\frac{\partial \theta}{\partial(\beta,\Theta)}$ is the Jacobian, showing how the generated parameters θ change when adjusting the QNN and mapping network parameters. $\nabla_\theta \mathcal{L}_{\text{CE}}$ is the gradient of the loss with respect to classical weights of AMC model. A detailed description of the backpropagation can be found in [23].

The computed gradients are propagated backward through QET, updating the QNN and mapping network parameters within the QPG, whereas the classical AMC is kept non-trainable.

3 Experimental Setup

3.1 Conventional AMC Model

In this study, we adopt the MC-Net model proposed in [6] as a case study for classical AMC, since it is both widely used and open-source. The MC-Net architecture is shown in Fig. 3. The model comprises 121611 trainable parameters.

3.2 Dataset

This study employs the RadioML 2016.10A dataset [8], a widely used dataset in AMC research. The dataset comprises eight digital and three analog modulation types. In total, 220000 signals were generated over 20 signal-to-noise ratio (SNR) levels spanning –20 dB to 18 dB, with 1k signals allocated per modulation per SNR. Each signal consists of 128 samples, which are represented as a

2×128 matrix, separating the real and imaginary components of the complex time samples.

3.3 Training Hyperparameter Configuration

All experiments were performed on a Dell Precision 3680 workstation equipped with an Intel® Core™ i7-14700 processor, 64 GB of RAM, a 2 TB SSD, and an NVIDIA GeForce RTX 4090 GPU. Training was conducted using the Adam optimizer with an initial learning rate of 0.001 and a batch size of 64. A maximum of 500 epochs was allowed, with early stopping applied if the validation loss did not improve for 30 consecutive epochs.

3.4 Evaluation Metrics

Accuracy is a key metric for assessing the classification performance of the DL-based AMC model. It represents the ratio of correctly predicted instances to the total number of instances in the test set. The accuracy can be expressed as

$$Acc = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \tag{12}$$

To evaluate the impact of QET relative to the traditional approach, we define the accuracy loss as

$$Acc_{Loss} = Acc_{baseline} - Acc_{QET}, \tag{13}$$

where Acc_{QET}, $Acc_{baseline}$ denote the accuracies of the DL-based AMC model trained using the QET framework and using the classical approach, respectively.

To evaluate computational efficiency, we consider parameter efficiency (PE), defined as

$$\Delta\mathcal{C}(\%) = \left(1 - \frac{\mathcal{C}_{\text{QET}}}{\mathcal{C}_{\text{AMC}}}\right) \times 100, \tag{14}$$

where $\mathcal{C}_{\text{QET}}$ is the number of trainable parameters in the QET framework and $\mathcal{C}_{\text{AMC}}$ is that of the baseline model.

In addition, we measure the average training time per epoch (T_e) in seconds, defined as

$$T_e = \frac{\text{Total training time}}{\text{Number of epochs}}, \tag{15}$$

where the total training time is recorded until model convergence.

4 Experimental Results

4.1 Performance of the AMC Model and QET Framework

Fig. 4 illustrates the classification accuracy of MC-Net across different SNR values. The MC-Net model trained using the conventional approach achieves an

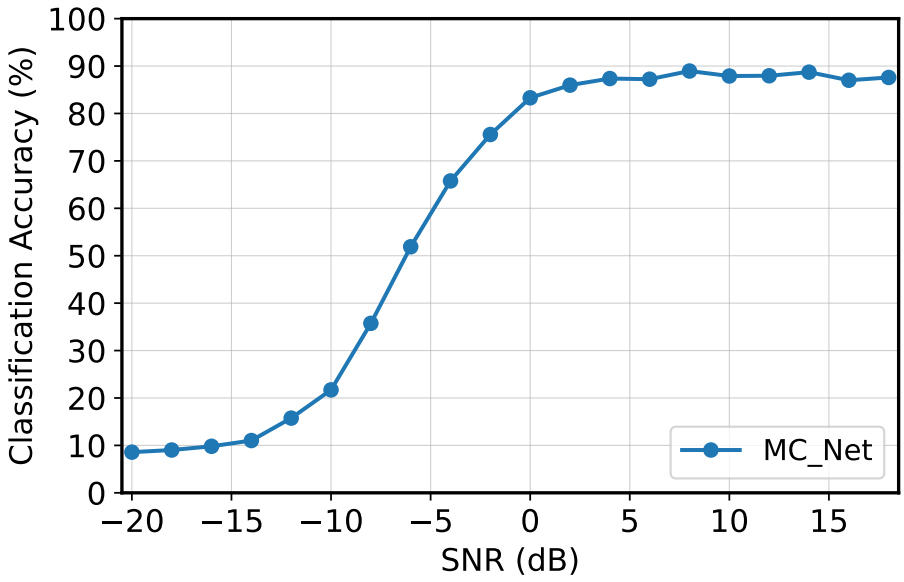

Fig. 4. Classification accuracy of MC-Net across different SNR values.

average accuracy of 58.8%, with performance rising to nearly 90% at high SNR levels.

Additionally, we evaluate the performance of the MC-Net model when trained using both the conventional approach and the QET framework. This work focuses on improving the training efficiency of the AMC system. To this end, we employed the QET framework and assessed its performance by varying the number of qubits from 5 to 9, denoted as QET-5 to QET-9, since QET provides parameter efficiency when the number of qubits is greater than or equal to 5.

As shown in Table 2, the QET framework demonstrates a substantial improvement in PE, expressed as $\Delta\mathcal{C}$ (%), compared to the classical MC-Net baseline. The baseline model requires 121,611 trainable parameters, whereas the QET framework dramatically reduces this number. For example, QET-5, implemented with 5 qubits, requires only 67,823 parameters to train the MC-Net model, which is a 44.23% reduction compared to the baseline. This reduction becomes more pronounced as the number of qubits grows. QET-6 and QET-7 require 35,440 and 20,185 parameters, respectively, while QET-9, contains only 10,526 parameters—a remarkable reduction of 91.34% compared to classical training methods. These results clearly demonstrate that the QET framework allows MC-Net to be trained with significantly fewer parameters, which is particularly advantageous for deployment in resource-constrained environments with limited memory budgets.

It can be observed that parameter reduction exhibits a saturation effect at higher Q values. The largest savings appear at lower qubit counts, as QET-

Table 2. Comparison of QET models of different Q with the classical MC-Net

Model	Parameters	$\Delta\mathcal{C}$ (%)	Acc (%)	Acc_{Loss} (%)	T_e
MC-Net	121,611	–	58.8	–	17.1
QET-5	67,823	44.23	59.1	−0.3	102.3
QET-6	35,440	70.85	59.4	−0.6	126.5
QET-7	20,185	83.40	57.7	1.1	146.3
QET-8	12,501	89.72	56.6	2.2	156.8
QET-9	10,526	91.34	55.7	3.1	145.2

5 reduces the parameter count from 121,611 to 67,823, while QET-6 further reduces it to 35,440, giving a difference of 32,383 parameters between the two models. Beyond this point the reductions become smaller, with QET-7 reaching 20,185 parameters, QET-8 reaching 12,501 with a difference of 7,684 compared to QET-7, and QET-9 reaching 10,526 with a difference of only 1,975 compared to QET-8. Moreover, the gains in PE come with an accuracy trade-off, as discussed in the sequel.

In terms of classification accuracy, the QET models achieve competitive performance relative to the classical MC-Net, with certain configurations even outperforming the baseline. For example, QET-5 and QET-6 achieve slightly higher accuracies of 59.1% and 59.4%, respectively, corresponding to negative accuracy losses of −0.3% and −0.6%. This indicates that QET outperforms the classical method. However, as the number of qubits continues to increase, accuracy gradually declines. For instance, QET-7 and QET-8 obtain accuracies of 57.7% and 56.6%, with corresponding accuracy losses of 1.1% and 2.2%. At the highest setting, QET-9 achieves only 55.7%, incurring a 3.1% loss. These results highlight a clear trade-off -while QET offers superior PE with higher qubit counts, the accompanying decline in accuracy arises because the mapping network becomes progressively smaller, leading to underfitting. Furthermore, while QET enables significant parameter reduction, QET-6 represents the most favorable trade-off between efficiency and accuracy, making it the optimal solution among the QET variants.

Despite the advantages in PE, the QET models exhibit a considerable increase in average training time per epoch compared to the baseline. The classical training method requires only 17.1 seconds per epoch to train the MC-Net model. In contrast, QET-5 requires 102.3 seconds, and the training time further increases as more qubits are incorporated into the framework, peaking at 156.8 seconds for QET-8. Even QET-9, despite utilizing the fewest parameters, still requires 145.2 seconds per epoch. This substantial rise in training time is primarily due to the fact that the QET framework is currently simulated on classical hardware using TorchQuantum, which imposes significant computational overhead and thus prolongs response time. Nevertheless, as quantum computing technology continues to mature, it is expected that the training time of the

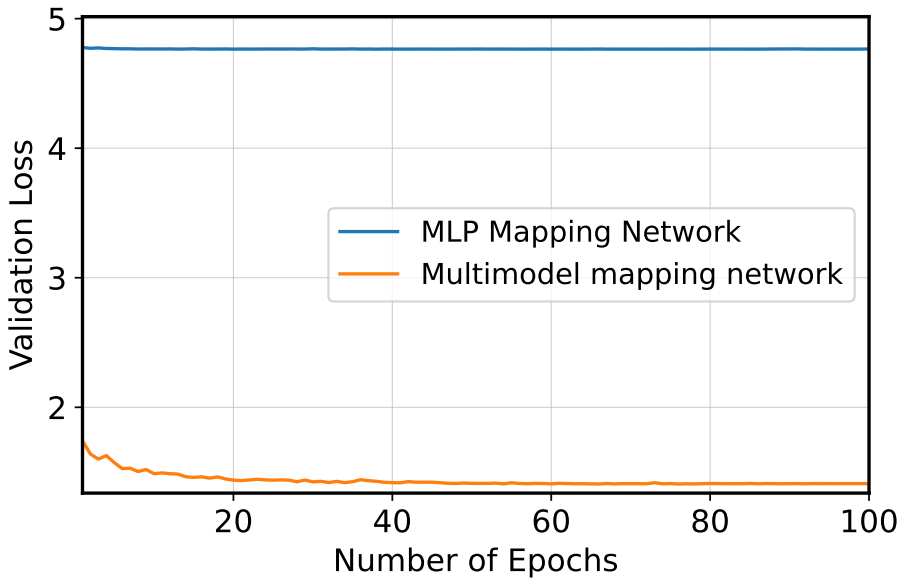

Fig. 5. Comparison of validation loss for QET models using MLP Mapping Network and Multimodel mapping network.

QET framework will be dramatically reduced when executed on actual quantum hardware.

4.2 Evaluation of the Proposed Multimodel mapping network

In this subsection, we evaluate the performance of the proposed mapping network by comparing Multimodel mapping network against conventional MLP-based mapping networks. Fig. 5 compares the convergence behavior of QET when employing either an MLP-based mapping network or the proposed Multimodel mapping network.

The QET framework struggles to converge with the MLP-based network, as indicated by the nearly constant validation loss throughout all epochs. This suggests that the MLP-based architecture suffers from the BP problem. To train an MC-Net model with 121,611 parameters, the QET-MLP framework requires 17 qubits, as shown in (9). At such a high qubit count, the BP effect becomes more pronounced, making optimization extremely difficult. This observation is consistent with the findings reported in [26].

In contrast, Multimodel mapping network allows proactive control of the qubit count by treating it as a tunable hyperparameter, thereby enhancing scalability and eliminating the need for an excessive number of qubits. In this experiment, QET was configured with 6 qubits. As illustrated in Fig. 5, the validation loss achieved by Multimodel mapping network is significantly lower than that of

the MLP-based mapping network, clearly demonstrating the superiority of the proposed approach and indicating that it effectively mitigates the BP problem.

5 Conclusion

In this paper, we have proposed the QET framework to address the challenge of training inefficiency in AMC systems. Specifically, QET integrates QNNs with Multimodel mapping network to generate parameters for classical AMC models. In addition, we have introduced a Multimodel mapping network that not only outperforms existing mapping approaches but also effectively mitigates the BP problem. Extensive experiments conducted on different QET variants (QET-5 to QET-9) have demonstrated that QET-6 provides the most favorable trade-off, achieving near-classical accuracy while maintaining high PE. Notably, QET-6 was able to reduce the number of trainable parameters by more than 70% compared to the baseline MC-Net and achieved superior classification accuracy over conventional training methods. These results highlight the potential of QET as a promising paradigm for enhancing the efficiency of AMC systems, especially in scenarios with constrained computational and storage resources.

Acknowledgments

This work was supported in part by the Cyber AI Hub Doctoral Training Program at the Centre for Secure Information Technologies (CSIT), Queen's University Belfast, supported by the UK Government as part of the New Deal for Northern Ireland, administered through Innovate UK/UKRI, and the UK Engineering and Physical Sciences Research Council (EPSRC), through the EPSRC Hub on All Spectrum Connectivity (EP/X040569/1 and EP/Y037197/1). The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, and in part by the NSERC CREATE program (Grant number 596205-2025).

References

1. Thien Huynh-The, Quoc-Viet Pham, Toan-Van Nguyen, Thanh Thi Nguyen, Rukhsana Ruby, Ming Zeng, and Dong-Seong Kim. Automatic modulation classification: A deep architecture survey. *IEEE Access*, 9:142950–142971, 2021.
2. Xiang Wang, Yurui Zhao, and Zhitao Huang. A survey of deep transfer learning in automatic modulation classification. *IEEE Transactions on Cognitive Communications and Networking*, 11(3):1357–1381, 2025.
3. Mingyang Du, Jifei Pan, and Daping Bi. A contrastive learner for automatic modulation classification. *IEEE Transactions on Wireless Communications*, 24(4):3575–3589, 2025.

4. To Truong An, Antonios Argyriou, Annisa Anggun Puspitasari, Simon L. Cotton, and Byung Moo Lee. Efficient automatic modulation classification for next-generation wireless networks. *IEEE Transactions on Green Communications and Networking*, 2025.
5. Rui Ding, Fuhui Zhou, Qihui Wu, Chao Dong, Zhu Han, and Octavia A. Dobre. Data and knowledge dual-driven automatic modulation classification for 6G wireless communications. *IEEE Transactions on Wireless Communications*, 23(5), 2024.
6. Thien Huynh-The, Cam-Hao Hua, Quoc-Viet Pham, and Dong-Seong Kim. MCNet: An efficient CNN architecture for robust automatic modulation classification. *IEEE Communications Letters*, 24(4):811–815, 2020.
7. Thien Huynh-The, Quoc-Viet Pham, Toan-Van Nguyen, Thanh Thi Nguyen, Daniel Benevides da Costa, and Dong-Seong Kim. Rannet: Learning residual-attention structure in CNNs for automatic modulation classification. *IEEE Wireless Communications Letters*, 11(6):1243–1247, 2022.
8. Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks: 17th International Conference, EANN 2016, Aberdeen, UK, September 2-5, 2016, Proceedings 17*, pages 213–226. Springer, 2016.
9. Hui Lu, Ruoliu Zhang, Shiqi Wang, and Yuhui Shi. Robustness assessment of DL-based automatic modulation classification model via channel-aware adversarial perturbation. *IEEE Transactions on Cognitive Communications and Networking*, 2025.
10. To Truong An and Byung Moo Lee. Robust automatic modulation classification in low signal to noise ratio. *IEEE Access*, 11:7860–7872, 2023.
11. Annisa Anggun Puspitasari, To Truong An, Mohammed H. Alsharif, and Byung Moo Lee. Emerging technologies for 6G communication networks: Machine learning approaches. *Sensors*, 23(18), 2023.
12. Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. TinyTL: reduce memory, not parameters for efficient on-device learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.
13. Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC20)*, pages 1–12, 2020.
14. Yao Lu, Yutao Zhu, Yuqi Li, Dongwei Xu, Yun Lin, Qi Xuan, and Xiaoniu Yang. A generic layer pruning method for signal modulation recognition deep learning models. *IEEE Transactions on Cognitive Communications and Networking*, 11(4):2123–2134, 2025.
15. Zhuangzhi Chen, Zhangwei Wang, Xuzhang Gao, Jinchao Zhou, Dongwei Xu, Shilian Zheng, Qi Xuan, and Xiaoniu Yang. Channel pruning method for signal modulation recognition deep learning models. *IEEE Transactions on Cognitive Communications and Networking*, 10(2):442–453, 2024.
16. Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1):2, 2020.
17. To Truong An, Simon L. Cotton, Junqing Zhang, Yuan Ding, and Trung Q. Duong. LoRa radio frequency fingerprinting identification using a hybrid quantum-classical neural network. In *Proc. IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, pages 1–6, 2024.

18. Muhammad Shohibul Ulum, Uman Khalid, Jason William Setiawan, Trung Q. Duong, Moe Z. Win, and Hyundong Shin. Variational anonymous quantum sensing. *IEEE Journal on Selected Areas in Communications*, 42(9):2275–2291, 2024.
19. Bao-Nhi Dang Tran, Muhammad Fahim, Bradley D. E. McNiven, Mohsen Guizani, Hyundong Shin, and Trung Q. Duong. Quantum LSTM model for estimation of energy expenditure in human aging using wearable iot healthcare technology. *IEEE Internet of Things Journal*, 12(22):46051–46064, 2025.
20. Bishmita Hazarika, Keshav Singh, Octavia A. Dobre, Chih-Peng Li, and Trung Q. Duong. Quantum-enhanced federated learning for metaverse-empowered vehicular networks. *IEEE Transactions on Communications*, 73(6):4168–4183, 2025.
21. Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature computational science*, 2(9):567–576, 2022.
22. Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
23. Chen-Yu Liu, En-Jui Kuo, Chu-Hsuan Abraham Lin, Jason Gemsun Young, Yeong-Jar Chang, Min-Hsiu Hsieh, and Hsi-Sheng Goan. Quantum-Train: rethinking hybrid quantum-classical machine learning in the model compression perspective. *Quantum Machine Intelligence*, 7(2):80, 2025.
24. Chen-Yu Liu, Chu-Hsuan Abraham Lin, Chao-Han Huck Yang, Kuan-Cheng Chen, and Min-Hsiu Hsieh. QTRL: Toward practical quantum reinforcement learning via quantum-train. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, pages 317–322, 2024.
25. Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3:010313, 2022.
26. Martín Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J Coles, Lukasz Cincio, Jarrod R McClean, Zoë Holmes, and M Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics*, 7:174–189, 2025.