Quantum Transfer Learning for Natural Language Classification: Alternative Circuit Designs and Evaluation

B240712

Master of Science

Computer Science

School of Informatics

University of Edinburgh

2024

Abstract

Quantum Transfer Learning, a variation of transfer learning in which a variational quantum circuit is used to to process the output of a pretained classical model, has received significant interest in recent years. To achieve QTL, Mari et al. 2020 proposed the Dressed Quantum Circuit architecture, including classical layers learning input and output strategies, to allow for simple connection between large classical circuits and small quantum circuits. Dressed Circuits have been demonstrated to allow quantum machine learning on a variety of problems, but most authors have made use of the same architecture proposed by Mari. The present paper investigates alternative implementations of the Dressed Quantum circuit, including alternate ansatz and embeddings. The circuits assessed include the previously known DQC, a circuit with the addition of RZ gates, variants making use of amplitude encoding and a variant with only encoding and measurement and no quantum gates. It concludes that the classical layers currently dominant performance of the Dressed Circuits, and suggests the use of encoding only circuits as benchmark.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*B240712*)

Acknowledgements

I would first like to thank my supervisor Dr. Petros Wallden for regular formative discussions during the course of the project, including suggestions for quantum circuits to implement. I would also like to thank Dr. Andrea Mari for making publicly available the code for the 2020 paper that first popularised Quantum Transfer Learning.

Table of Contents

Chapter 1

Introduction

Quantum Machine Learning is an area of great interest in recent years, offering both theoretical speedups over classical machine [1] and greater expressive power than comparable classical networks [2]. Of particular interest, Variational Quantum Algorithms (VQA), in which the quantum circuits are controlled by trainable classical parameters, may be practically implemented on current Noisy Intermediate Scale Quantum systems (NISQ). Though lacking theoretical proofs of supremacy, VQAs have already been shown to allow for effective quantum processing [3]. However, current quantum circuits are strictly limited in size, due to noise and limited coherence times, and so are impractical for processing large data.

Quantum Transfer Learning (QTL), proposed by Mari et al. [4, 5], aims to resolve this issue by adapting the technique of transfer learning; a classical technique allowing efficient fine-tuning of large pretrained networks for narrow tasks. Conventionally, most layers of a model are frozen, with only a small number of output layers being trained for a particular task. QTL utilises the frozen classical network to extract a small set of features from large input data to produce, before processing using a VQA.

To this end, [5] suggests a Dressed Quantum Circuit, in which the quantum circuit is trained at the same time as small classical networks for encoding to, and reading from, the quantum circuit. This allows for small smaller quantum networks to be easily connected to large classical networks, with the input network acting to reduce the number of input features while also learning an optimal encoding strategy.

QTL has proved popular, being applied to task ranging from medical diagnosis [6, 7, 8, 9, 10], Wi-Fi sensing [11] and optical character recognition [12]. The majority of the preceding work has applied QTL for a variation of image classification, as was the original work by Mari et. [5]. Less work has been conducted on applying QTL to the

the field of Natural Language Processing [13, 14].

While several papers have reported successful applications of QTL, there has been no investigation of how performance varies according to quantum circuit design; such as the impact of circuit width or qubit number, alternative ansatz or encoding. Additionally, there is little consideration of ascribing credit; determining whether training performance is due to the dressing layers or quantum circuit itself. While the originators of the Dressed Quantum Circuit claimed the quantum circuit carries out the essential processing, the impact of the classical layers has not previously assessed.

The following project first adapts the Dressed Quantum Circuit of Mari 2020 to fine tune a variant of BERT for equivalence recognition using the Microsoft Research Paraphrase Corpus (MRPC). We evaluate the training performance dependant on quantum circuit width, and aim to extrapolate performance to currently impractical, large circuits. We further evaluate performance of the circuit with alternative ansatz and methods of encoding. Finally, we propose a novel method of benchmarking quantum transfer learning by applying a quantum circuit with no active gates to ascertain the impact of classical layers alone. We re-evaluate the classifier of Mari 2020 accordingly. Overall, we aim to determine if quantum transfer learning has viable applications in the near term on requires significant development to have utility.

The present project aims to investigate the impact of quantum circuit design on the performance of QTL, and does not aim to produce an improved classifier. Similarly, no new quantum circuit designs or methods of encoding are suggested. The novel contribution of this project lies in evaluation of quantum transfer learning for numerous circuit architectures, and the provision of methods of benchmarking future projects.

In the following chapters: chapter 2 sets out background information; chapter 3 sets out the details of the hybrid classical to quantum classifier and training process; chapter 4 contains and discusses results from the training process; and chapter 5 provides interpretation and broader discussion of results. Finally, chapter 6 summarises results, considers the limitations of the present process and suggests future work.

Chapter 2

Background

2.1 Transfer Learning

Transfer Learning, TL, is a technique allowing for efficient reuse of pretrained networks for specific tasks. Cutting edge networks are often trained for general performance on large datasets, requiring long and expensive training to be applied to a new task. Alternatively, training a network from scratch on a narrow task is often impractical due to the limited amount of relevant data. TL aims to resolve both issues by fine-tuning only portion of a model on a specific task. As summarised by [5]:

1. Start with network A trained on dataset D_A for task T_A
2. Remove at least one layer of A to give reduced network A'.
3. Replace the removed layers with a new network B
4. While keeping weights in A' frozen, train B on a new dataset D_B for new task T_B

The truncated network A' acts may be viewed as a feature extractor, converting inputs into relevant features but not yet processing them. TL allows for high performance on specific tasks with low training times with lower computational demands.

2.2 Classical to Quantum Transfer Learning

The present paper is an investigation of techniques first introduced in Mari et al. "Transfer learning in hybrid classical-quantum neural networks" (2020) [5]. The paper considers several versions of quantum transfer learning (QTL), of which Classical to Quantum (C2Q) TL shows the most promise in the near term. C2Q-TL allows the use of

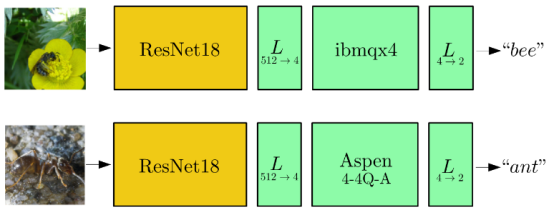

Figure 2.1: Operation of the hybrid classical to quantum circuit of Mari 2020. High resolution images are processed by ResNet18, giving 512 features. A linear layer reduces the features to four, which are processed on a VQC to give a binary prediction. Taken from Figure 4 of Mari 2020 [5]

successful classical models to produce a small number of information rich features for processing in small quantum circuits, potentially benefiting from the greater expressive power of quantum circuits [2]. For image process this can reduce the scale of the inputs from thousands or millions of bytes to hundreds.

Mari provided a hybrid classifier for binary image classification (ants/bees, dogs/cats, planes/cars). Images were processed by pretrained model ResNet18 to give a 512 feature "embedding" which were then input into a Dressed Quantum Circuit. The DQC reduced 512 input features to four, executed a four qubit quantum circuit and further reduced the four outputs to two class scores for a final prediction. Mari found a maximum accuracy of 96.7% using a simulated quantum circuit, with 95%
80% on the ibmqx4
Aspen-4-4Q-A quantum processors respectively.

2.2.1 Dressed Quantum Circuits

Introduced to allow for simple connection of quantum circuits with classical networks of arbitrary size, a Dressed Quantum Circuit surrounds a Variational Quantum Circuit (VQC) with two classical layers: an input layer for reducing the scale of inputs; and an output layer for converting measurements to a final output.

$$Q_D = L_{N_{in}\to N_q} \circ Q \circ L_{N_q\to 2} \tag{2.1}$$

where Q is a VQC, N_{in} is number of input features, N_q is the number of qubits.

The authors of [5] propose that during training the VQC learns to conduct the relevant processing while the classical layers learn optimal embeddings and readouts. It is a primary objective of the the present paper to investigate these claims.

2.3 Technical Background

Here we briefly establish the technical background of our methodology. As the techniques discussed are well known in the art, we will be not explain each in detail.

2.3.1 Variational Quantum Circuits

Variational Quantum Circuits (VQC) consist of a series of parameterised quantum operations (”gates”) for applying a learned unitary operation. VQC are initialised with a chosen basic structure, referred to as an ”ansatz” (by analogy to trial solutions). Control parameters are then trained using conventional optimisation [15].

At present, many VQA are constructed using ”hardware-efficient ansatz”, designed for practical implementation on real quantum hardware. Efficient ansatz are repeated layers of single qubit rotation gates followed by two qubit entangling operations [16]. The entangling operation is commonly a set of Controlled X gates, either applied between every pair of qubits (each-to-each) or between neighbouring pairs only (nearest neighbours)[1] [3]. The specific choice of circuit structure has significant impact on the performance of the variational circuit. More complex ansatz, such as multiple rotations per layer or longer range entanglement, allow for greater expressive power, but also increase noise and runtime of the circuit.[17, 18].

Classical information is embedded, or encoded, into VQC in a number of ways, discussed in section 3.3. Results are extracted from a quantum circuit through measurement of an observable, most often the expectation value of a Pauli operator.

”Quantum Neural Networks”, deep variational circuits, have been applied to simple classification tasks, achieving worse performance than a comparable classical circuits but suggesting the possibility of future improvements [19]. Abbas 2021 argues that QNN have theoretical benefits over classical networks, due to less flat loss landscapes (shown by less concentrated Fischer Information Spectra) [2].

2.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) is currently a one of the dominant models in Natural Language Processing. BERT is a high performing architecture based on deep pre-training of transformers, in which tokens are interpreted

[1] While each-to-each provides the maximum entanglement, nearest neighbour is often more practical on current hardware.

with both left and right contexts. Pre-training is carried out using a Masked Language Model (MLM), the network is trained to reconstruct sentences with words randomly removed/masked; and Next Sentence Prediction (NSP). Models are trained on large unsupervised datasets, followed by task specific supervised fine-tuning. BERT was initially trained using the Google bookcorpus and Wikipedia.

BERT was designed to allow for classification using a single output layer, with no need for complicated task specific architecture. Additionally, bidirectional self-attention allows for multiple inputs to be encoded into a single set of tokens, simplifying fine tuning on comparison tasks. When first published, BERT showed record beating performance on all GLUE tasks (discussed below).

The base BERT model comprises 12 hidden layers of 768 nodes and 12 attention heads, for a total of 110 Million parameters. Conventionally, all layers of the network are fine tuned for a particular task.

The transformer architecture is based on self-attention, a weighting for each token in the sentence. Inputs are first tokenized into numerical strings. Transformers are able to use multiple attention heads in parallel, allowing for parallelisation of the models and significant speed-ups, particularly when used with GPUs. For details see [20].

2.4.1 TinyBERT

A reduced size version of BERT-base generated using a Knowledge Distillation KD method to reduce the number of layers and parameters. KD perform distillation during both pretraining and fine tuning to maintain generalist performance. TinyBERT consists of two layers, each of 128 nodes with two attention heads, giving 4.4 million parameters compared to BERT-base with 110.1 million. For certain tasks, TinyBERT can run 65 times faster while retaining 91% of the performance of BERT-base [21].

2.4.2 Optimisers

As VQC are controlled by classical parameters, and as the unitary operations they implement are differentiable, they may be trained using classical gradient descent.

2.4.2.0.1 Adam A currently ubiquitous optimiser, Adam is a variation of stochastic gradient descent (SGD) that performs learning rate adaptation for each parameter using from estimates of the first and second moments of the gradient [22]. Adam shows good performance with high dimensional data, or features with sparse gradients. Additionally,

Adam is simple to implement and memory efficient, scaling linearly with the number of parameters (rather than quadratically as is generally required to implement Natural Gradient Descent). The currently most common version of Adam includes includes weight decay, a widely used version of regularization, which has been shown to improve generalisation [23].

Adam is effective for both for classical and quantum training and is provided in both PyTorch or Pennylane.

Hwang 2024 [24] has recently considered the connection between Adam and Natural Gradient Descent (NGD), which adjusts learning rates based on the curvature of the loss landscape calculated using the Fischer Information Matrix. Hwang sets out that Adam is an approximation of NGD, and proposes several improvements accordingly [2].

2.4.2.0.2 Rotosolve We briefly note that there exist specialised optimisers for quantum circuits. Rotosolve operates by individually changing the angle, and optionally basis, for each gate.

While this shows good performance for small networks, Adam shows better performance for larger number of parameters and is able to train classical and quantum networks equivalently [3].

2.4.3 GLUE

General Language Understanding Evaluation (GLUE) consists of a series of nine benchmark tests provided for a complete evaluation of natural language models, associated datasets for each, and a public leaderboard of performance. For full evaluation, a model is finetuned for each GLUE task, evaluated on the test set and the individual results combined to give an overall score [25]. [4]

The present paper only trains for MRPC, with CoLA and SST2 used for early investigations.

2.4.3.0.1 MRPC The Microsoft Research Paraphrase Corpus, MRPC, contains 5801 sentences pairs extracted from online news, manually labelled with whether or not they have equivalent semantic meanings. The dataset is split into 4076 training pairs (further divided into 3668 training pairs and 408 validation pairs), and 1725 testing pairs. [27]

[2] Available from `https://github.com/lessw2020/FAdam_PyTorch/tree/main`

[3] `https://pennylane.ai/blog/2022/06/how-to-choose-your-optimizer/`

[4] Due to the recent rapid development of Natural Language Processing, a new set of more difficult baselines have been proposed under the name of SuperGLUE [26]

2.4.3.0.2 CoLA A set of 10657 sentences drawn from linguistics publications, labelled for grammatical acceptability (whether the sentence is grammatically correct) by expert authors. [28]

2.4.3.0.3 SST-2 A set of 11855 sentences extracted from film reviews, annotated with sentiment by human judges. SST-2 (SST binary) is annotated as positive or negative[29].

2.4.4 Related Literature

QTL has been applied to tasks ranging from medical diagnosis [6, 7, 8, 9, 10], Wi-Fi sensing [11], optical character recognition [12], and identifying recyclables [30]. The majority of the preceding work adapted QTL for a version of image classification, continuing the original work by Mari et al [5].

Recently several authors have applied QTL with Natural Language Processing. Most relevant for the present work are Buonaiuto et al. 2024, [13] who investigated quantum transfer learning to determine if inputs are viable Italian. They found the quantum transfer learning paradigm to be competitive with classical training methods, and qualitatively argued that the quantum methods could classify more complex sentences. The paper considered dressed quantum circuits used applied as the classification head of BERT and ELECTRA, both current high performing transformer networks. Unlike Mari, Buonaituo encoded input features into the quantum variational circuit using amplitude encoding. Implementations using BERT and ELECTRA were found to have performance competitive with classical models, with BERT-Quantum performing slightly worse than BERT-Classical and ELECTRA-Quantum performing worse on some metrics and better on another.

Also relevant is Ardeshir-Larijani & Fatmehsari 2024, [14] in which the authors apply quantum transfer learning for Natural Language Processing for spam identification, which made use of BERT-Large, and a five qubits VQC. The network was trained on a dataset of text messages labelled as spam or not spam, finding comparable performance to classical classifiers.

Note that while previous papers have focused on the performance of the classifiers resulting from the transfer learning process, the present paper aims to investigate the impact of variants of the quantum circuit itself.

Chapter 3

Methodology

Here we describe the relevant training problem, the classifiers to be assessed, and other aspects of experimental design.

3.1 MRPC

To allow for simple evaluation and comparison, this project is trained on an NLP industry standard benchmark. As the present project aims to investigate a number of circuits with differing qubits numbers, the dataset and problem were selected to allow for practically short training times. Of particular importance was the size of the training set, as large quantum circuits require significant running times for each datapoint.

To this end, the classifier was trained on the Microsoft Research Paraphase Corpus (MRPC), provided as part of GLUE (For details, see 2.4.3.0.1). MRPC comprises 5801 sentences pairs, labeled as "equivalent" or "not equivalent". When used as part of GLUE the dataset is divided into 3668 training pairs, 408 validation pairs and 1725 testing pairs. Results on MRPC are reported as Loss, Accuracy and F1, the harmonic mean of precision and recall. When training a bidirectional model such as BERT, both sentences are tokenised into a single input containing a special "separator" token, allowing for use of a conventional classifier architecture.

MRPC was chosen due to the small training set and the ability to use smaller versions of BERT show reasonable performance on the task. TinyBERT[1] [21] allows up to Accuracy 71.1% F1 81.1% on MRPC, in comparison to BERT-base which allows for 85.1%/89.3%.

[1] `https://huggingface.co/google/bert_uncased_L-2_H-128_A-2`

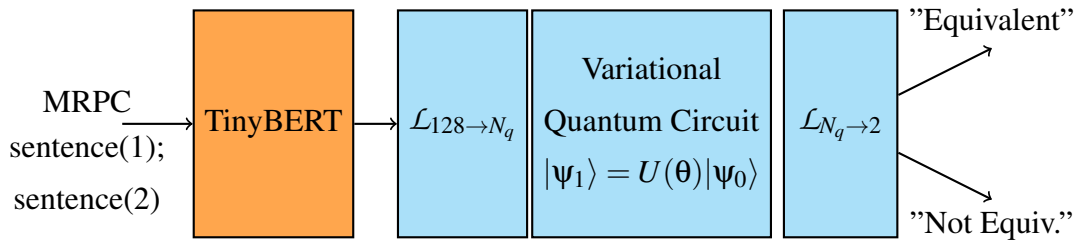

Figure 3.1: Operation of the hybrid classical to quantum classifier. MRPC sentence pairs are tokenized, processed by TinyBert to produce a 128 feature embedding, down-projected to reduce from 128 qubits to the number of qubits (N_q). The reduced features are embedded in a VQC which applies a learned unitary operation. Outputs are down-projected to give an output prediction. During training, all parameters in the TinyBERT network are frozen.

3.1.1 Balancing Dataset

The MRPC dataset is unbalanced, with 68% positive labels. During initial testing, it was observed that all training runs resulted in similar performance, with accuracy converging to near to the positive label percentage. To clearly accentuate differences in training performance, we created a balanced version of MRPC. The balanced training set contains 2392 sentence pairs with equal proportion positive and negative. Balanced versions of the validation set and test set were also constructed.

While this allows for clearer assessment of training performance, it prevents comparison to state of the art models or the GLUE leaderboard.

3.1.2 Other datasets considered

Early investigations were conducted on the CoLA dataset. However, due to the difficulty of the problem training could not be accomplished using smaller versions of BERT. In additional, the large size of the training set made gathering significant volumes of data impractical.

Further testing considered SST2. While it allows for a smaller classical model, the large training set resulted in long training for high qubit circuits.

3.2 Classifier Structure and Operation

Here we describe the complete classical to quantum transfer learning pipeline.

3.2.1 Classical Processing

3.2.1.1 Data Pre-processing

The MRPC dataset is loaded and normalised. For each datapoint, sentences are concatenated and tokenised into a vector of integers.

3.2.1.2 Classical Model: TinyBERT

As this project is concerned with the performance of the final classifier head, most training will use the smallest model practical for the task. For comparison, small batches of training are carried out using a larger model.

Datapoints are processed by a pretrained TinyBERT model (see section 2.4.1, which outputs an embedding, $\mathbf{x}$, comprising 128 real numbers. During training, the weights of the BERT network are not updated.

3.2.2 Dressed Quantum Network

Embeddings are input into a Dressed Quantum Circuit, consisting of a classical input layer, a variational quantum circuit and a classical output layer.

3.2.2.1 Input Layer

The input layer is a classical linear layer which acts to reduce the dimensionality of the input to one appropriate for encoding into the VQC. The output is passed into a hyperbolic tangent activation to ensure valuables are angles in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$:

$$\mathcal{L}_{128\to N_f} : \mathbf{x} \to \mathbf{y} = \frac{\pi}{2}\tanh(\mathrm{W_i}\mathbf{x} + \mathbf{b_i}) \tag{3.1}$$

where W_i is a matrix of edge weights, b_i is a vector of biases and N_f is the number of features accepted by the following circuit.

During training, the input layer learns both the optimal encoding both for dimensionality reduction and embedding into the quantum circuit. The input layer is included in all embodiments save for 3.4.3.2.

3.2.2.2 Embedding

The reduced size output are then be used to generate a valid quantum state:

$$\mathcal{E} : \mathbf{y} \to |\psi_y\rangle \tag{3.2}$$

Multiple methods of encoding are used in the art, of which this paper considers two: Angle encoding and Amplitude Encoding. For full discussion see section 3.3.

Angle encoding embeds each feature as the angle of a single qubit Pauli rotation gate, requiring a qubit for each input feature: $N_q = N_f$, where N_q is qubit number.

Amplitude encoding embeds features as the amplitudes of multi-qubit states. As the number of multi-qubit states scales exponentially with the number of qubits, amplitude encoding requires $N_q = \log_2(N_f)$.

3.2.2.3 Variational Quantum Circuit

The VQC applies six layers of operations, each layer consisting of parameterised, single qubit Pauli rotation gates and an entanglement operation consisting of nearest neighbour controlled X gates. Nearest neighbour entanglement was chosen for simplicity of implementation on actual hardware.

In this paper we consider three ansatz: RY & Controlled X gates only; RY, RZ & Controlled X gates; and encoding only (encoding immediately followed by measurement), discussed below. Training updates the relevant angle parameters, but does not otherwise change the ansatz.

Data is extracted from the quantum circuit by measuring the expectation of the Pauli Z operator for each qubit:

$$\mathbf{z}_i = \langle \psi | Z_i | \psi \rangle \tag{3.3}$$

where z is a vector of real numbers in the range [-1,1], and Z_i is Pauli Z acting on qubit i. The number of output features is equal to the number of qubits, regardless of the encoding or ansatz.

3.2.2.4 Output Layer

The output layer is a classical linear reduction to two classes:

$$\mathcal{L}_{N_q \to 2} : \mathbf{z} \to \mathbf{s} = \mathrm{W_o}\mathbf{x} + \mathbf{b_o} \tag{3.4}$$

where s is a two feature vector containing a score for each class, W_o & b_o are output weights and biases. Finally, the output layer is used to generate a prediction by selecting the class with the highest score: $\mathrm{pred} = \mathrm{argmax}_i(s_i)$

3.2.3 Model Training and Evaluation

The hybrid classifier is trained using the labelled MRPC dataset, classifying datapoints according to whether the two sentences are equivalent. The model is trained to minimise Cross Entropy Loss between the output scores and data labels, updating the parameters of the DQC while keeping parameters of BERT frozen. Each epoch the model is also evaluated on a validation set. When training is completed, the model with best performance on the validation set is selected and evaluated on a held out training set, computing Loss, Accuracy and F1 scores.

3.3 Quantum Encoding

There are multiple methods of encoding classical data into a quantum circuit. In current NISQ devices encoding methods need to compromise between representing data accurately and reducing the number of gates and qubits needed to encode the data. Two methods will be assessed in the present work.

3.3.1 Angle Encoding

Angle encoding represents each feature as the rotation of a separate qubit about the Bloch Sphere. Input data is normalised to a valid angle, and applied as such by a Pauli rotation gate. For example, when encoding using rotations about the Pauli Y axis, qubits are initialised as $|0\rangle$, converted to $|+\rangle$ by a first layer of Hadamard gates, after which each qubit is rotated by a single qubit R_y gate according to an input feature:

$$\mathcal{E}(\mathbf{x}) = \bigotimes_{k=1}^{N_q} \left(R_y \left(x_k \frac{\pi}{2} \right) H |0\rangle \right)$$

The resulting state is not entangled; as such angle encoding is also known as product encoding [31]. Additional entangling operations are required following embedding to enable general quantum states. Angle encoding is commonly used due to the high speed and efficiency, requiring only two gates for each input feature. For full encoding one qubit is needed per input feature ($N_q = N_f$). Pennylane provides an embedding function for each Pauli axis.

This method is used by Mari 2020 and the majority of later work on quantum transfer learning. On current systems the number of qubits is much smaller than the output of even small networks, and so input features must be down-projected prior

to encoding. It was this requirement that motivated the development of the Dressed Quantum Circuit introduced by Mari [5].

3.3.2 Amplitude Encoding

Amplitude encoding makes use of quantum superposition to allow for an exponentially large amounts of data to be input into a circuit. Input features are encoded as amplitudes of multi-qubit quantum states in a superposition of states: [31]

$$\mathcal{E}(\mathbf{x}) = |\psi_x\rangle = \sum_{i=1}^{N} x_i |i\rangle,$$

As the number of possible states scales exponentially with the number of qubits, Amp. Enc. requires $N_q = \log_2(N_f)$ for an accurate encoding. This provides the benefit of exponential storage of the input features, allowing for processing of large classical feature-sets in practical scale quantum devices. For example, the output of $BERT_base$ can be encoded into 10 qubits, while TinyBert can be encoded into 7. This also results in exponentially reduced training times for the same number of features. Notably for our purposes, ammplitude encoding allows for the removal of the down-projection layer of the DQC, discussed below.

However, amplitude encoding is implementationally and computationally complex, in general requiring 2^{N_f} rotation and CNOT gates to accomplish. [32, 33] In current noisy quantum circuits, this is likely to remove any advantage the method may have provided. Amplitude encoding also requires normalisation of the input feature vector.

Preliminary testing was conducted using amplitude embedding, finding extremely long runtimes for moderate qubit numbers. In addition, the method showed extreme numerical unstability for high qubit number, requiring small batch size and low learning rates to avoid diverging gradients in the classical layers of the dressed circuit.

3.3.2.1 Pennylane Simulated Amplitude Encoding

The Pennylane simulator provides for simulated amplitude encoding, by directly initialising the simulator in the required state[2],[3]. This allows for testing of the impact of amp. encoding with vastly reduced runtimes. The simulated amp. encoding was used to investigate additional implementations of DQC. [4]

[2] see documentation `https://github.com/PennyLaneAI/pennylane/blob/master/pennylane/templates/embeddings/amplitude.py`

[3] When running on hardware, the method instead implements Möttönen state preparation.[32]

[4] Note that while provided by Pennylane, this method is non-differentiable, see `https://docs.pennylane.ai/en/stable/code/api/pennylane.AmplitudeEmbedding.html`.

Name	Type	Gates	Encoding	Input Layer	Parameters
Classical	Classical	-	-	Yes	$130N_q$
RY Only	Quantum	RY + CNOT	Angle (RY)	Yes	$136N_q$
RY+RZ	Quantum	RY + RZ + CNOT	Angle (RY+RZ)	Yes	$270N_q$
Amp.Enc (Dressed)	Quantum	RY + CNOT	Amplitude	Yes	$128(2^{N_q}) + 8N_q$
Amp.Enc (Undressed)	Quantum	RY + CNOT	Amplitude	No	$8N_q$
Encoding only	Quantum	-	Angle (RY)	Yes	$130N_q$

Table 3.1: Summary of assessed circuits, showing combinations of types, gates used, encoding and presence of down-projection input layers. Name is how the circuit will be referenced in the following sections

Amplitude encoding has previously been used with QTL by Buonaiuto (2024) [13]

3.4 Circuit Variants

We now establish the variant circuits to be assessed:

3.4.1 RY only

The most simple circuit considered here; features $\{x_i\}$ are entered into the circuit via angle embedding (see 3.3). Processing is performed by 6 layers of parameterized RY, with entangling via nearest neighbour CNOT.

$$|\psi_l\rangle = U(\mathbf{w})|\psi_{l-1}\rangle \ = K\left(\bigotimes_{k=1}^{N_q} RY\left(w_k\right)\right)|\psi_{l-1}\rangle \tag{3.5}$$

where ψ_l is the state after layer l, $RY(\theta)$ is a rotation about the Pauli Y axis by θ radians, K is a nearest neighbour entangling operation, and w is a vector of classical parameters. Classical data is extracted by measuring Pauli Z for each qubit. Only w is updated during training.

This circuit was previously utilised by Mari 2020 [5] and much of the work developing on such [8, 9, 11], While known to be functional, the simple nature of the gates results in comparatively low hypothesis space explored by the circuit, making it likely to lack the benefits potentially offered by quantum machine learning [15].

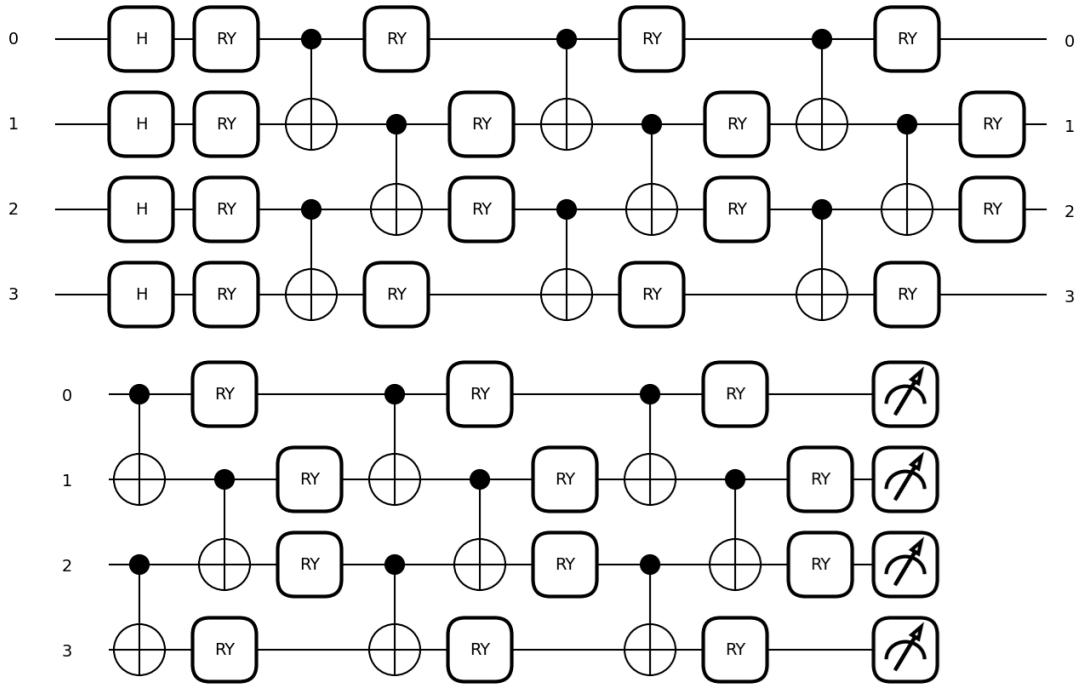

Figure 3.2: Schematic of a six layer, four qubit circuit using only RY and CNOT gates. Initial layer of RY gates is used for angle embedding. Later layer angles are trained variational parameters. Outputs are Pauli Z measurements.

3.4.2 RY + RZ

The second circuit is a variant of the circuit above with increased complexity, replacing all RY gates with RY followed by RZ, a rotation around the Pauli Z axis. Features are embedded using modified angle encoding, making use of both RY and RZ. Note that RX gates are not used as they commute with the entangling operation. Again, data is extracted by measuring Z for each qubit.

The use of two different rotation bases significantly increases the expressiveness of the variational circuit [18], potentially allowing for greater training performance over the RY only circuit above.

3.4.3 Amplitude Encoding

A variant of the above RY circuit; features are input via amplitude embedding and processed by six layers of single qubit RY and two qubit CNOT. While the processing layers of the variational circuit are equivalent to circuit 3.4.1, amplitude embedding allows for exponentially more data to be input. With 128 input features, seven qubits allows for embedding of all features into the circuit, only requiring normalisation.

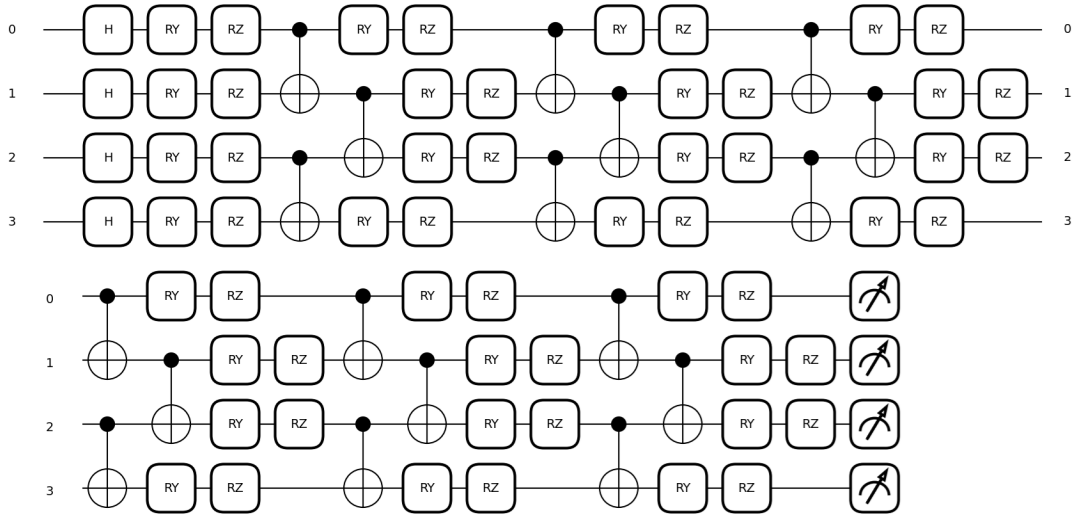

Figure 3.3: Schematic of a six layer, four qubit circuit using RY, RZ and CNOT gates. Initial layer of RY and RZ gates are used for angle embedding. Later layer angles are trained variational parameters. RX gates are not used due to commuting with the entangling operations (CX). Outputs are Pauli Z measurements.

Regardless of encoding, each qubit only results in a single output feature, and so the circuit necessarily acts to reduce the dimensionality of data.

Two versions of the amplitude encoding circuit are provided:

3.4.3.1 Amplitude encoding with input layer (Dressed)

The first implementation maintains all features of the Dressed Quantum Circuit previously described, comprising an input layer that projects from 128 features to 2^{N_q}. This allows for the simple use of the quantum circuit with any number of qubits. For fewer than seven qubits, the classical circuits acts to down-project the inputs, while for more than seven the input circuit is over-parameterised.

3.4.3.2 Amplitude encoding without input layer (Undressed)

As amplitude encoding allows for full encoding of information with a practically small number of qubits, the DQC can be implemented without using a classical down-projection layer. Output embeddings from BERT are normalised and encoded directly into the quantum circuit. This method requires at least $log_2(N_f)$. Of particular importance, this circuit allows investigation of the expressive power of the input layer.

3.4.4 Encoding Only / ”No Gates”

In previous analyses of Dressed Quantum Circuits, the input and output classical layers have been treated as efficient means for connecting a quantum circuit to classical inputs and outputs; assuming the input layer learns an optimal embedding, the quantum circuit performing the relevant processing and the output layer learns post-processing [5]. However, it can be observed that the input and output layers alone provide for a viable classification head, the training of which may be providing the dominant effect. Notably, for a 128 feature input, the input layer will contain $128N_q$ parameters, while the quantum circuit itself will be controlled by $6N_q$.

To investigate this possibility, we will be assessing a quantum circuit with no active quantum layers. All RY gates are set to zero (or ”frozen”) and no entangling layer was applied. Features were encoded into the quantum network using RY angle encoding, then immediately measured with no processing. The only training is performed on the input and output classical layers.

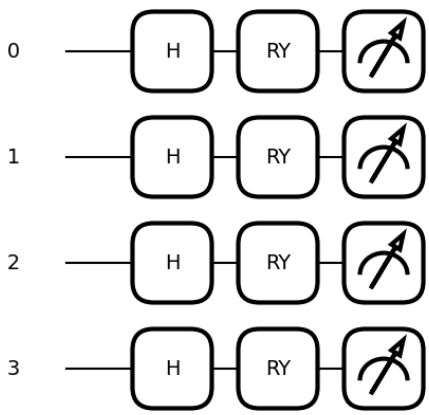

Figure 3.4: Schematic of a four qubit circuit with angle encoding only. Input features are embedded as angles of RY gates. Outputs are Pauli Z measurements.

This circuit is mathematically equivalent to applying $z = -\sin\left(\left(\frac{\pi}{2}\right)\tanh(x)\right)^5$. However, it is unlikely to be computationally equivalent, and so experiments will implement the encoding only circuit as a quantum node.

3.4.5 Classical

Evaluated to provide a point of comparison, the classical network connects the input and output layers of the dressed circuit directly, resulting in a two layer neural network with N_q hidden nodes. The network applies a $\mathrm{ReLU}(x) = \max(x, 0)$ activation. Note that for simplicity of reporting data, we will be referring to the number of intermediate nodes as ”number of qubits” N_q in future tables.

[5] See appendix D

3.5 Qubit Numbers

The present paper intends to investigate how performance of QTL scales with changes in qubit number. While early studies demonstrated the viability of small dressed circuits, higher qubit numbers will both allow for more expressive circuits, for information to be embedded with lesser reduction of dimensionality.

3.6 Training and Evaluation Details

The model was trained according to the process described in 3.2.3. Each model was trained at least six times for 30 epochs for each qubit number in the range 1-16, with the exception of circuit 3.4.3.2, which can only be trained for qubit numbers greater than 7. Additionally, RY only, Classical and Encoding only are evaluated for 17-18 qubits, due to their simplicity allowing for practical training times. All training was carried out using the Pennylane simulator[6]

3.7 Re-evaluation of Mari 2020

During the project, it became apparent there is limited, if any benefit, from applying a Variational Quantum Circuit with BERT for training on MRPC. However, this opens the question of whether QTL is of limited use for this specific architecture and problem, or if the results can be generalised. To this end, we investigate the performance of the our alternative circuits on the the image classification problem presented by Mari et al [5]. The classifier of Mari was adapted to use the alternative Variational Circuits introduced above. Of particular interest, the classification of ants bees was evaluated using a quantum network with no processing gates.

This re-evaluation provides the benefits of investigating a problem that is known to be solvable by a simple Dressed Quantum Circuit, allowing us to determine if QTL performance is being limited by other factors.

All experiments were repeated once, using the default parameters set out by Mari 2020 [5] (learning rate 4e-4, batch size 4, 30 epochs).

[6] `https://docs.pennylane.ai/en/stable/code/api/pennylane.device.html`

3.8 Finetuned BERTbase

To determined if the observed effects were a result of an inappropriately chosen base network, small scale comparison experiments were performed using a fine tuned version of BERT-base. While this does not explore the feasibility of transfer learning, it does address concerns over the use of an inappropriately weak classical network.

3.9 Choice of Hyperparameters

Brief initial experiments were conducted to determine the highest performing hyperparameters. No formal hyperparameter search was conducted.

- Optimiser: Adam with Weight decay, with default parameters.
- Learning Rate: 0.001. Higher rates were consistently found to offer give lower final loss.
- Scheduler: Learning rate reduced by factor of 0.1 every 10 epochs
- Batch size: 4
- Circuit Depth: 6 layers were chosen as a compromise between processing ability and training times.
- Epochs: Early data was gathered for 20 training epochs, as training was observed to plateu for longer training times. For full investigation, experiments were repeated training for 30 epochs.

Training was conducted on two devices: the Eddie CPU cluster, running on the Processor Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, 16GB virtual memory[7]; Laptop containing 12th Gen Intel(R) Core(TM) i7-1255U @ 3.2GHz, 64GB RAM.

3.10 Program Details

The transfer learning system was programmed in Python, adapted from the iPython notebook and tutorial provided by the authors of Mari 2020 [5, 4] [8]. The core training loop was left unchanged, as were the RY only DQC and quantum circuit functions.

[7] GPU clustering was not used to high levels of congestion on GPU nodes
[8] accessible from https://github.com/XanaduAI/quantum-transfer-learning

Alternative variants of the DQC were newly implemented for this paper. GLUE dataset loading and tokenization adapted from Notebook provided by github user "sikfeng" [9]

3.10.1 Relevant Packages

3.10.1.0.1 PyTorch Package providing neural network functionality with autodifferentiation. PyTorch maintains a record of the relevant gradient associated with each function during calculation, allowing easy backpropagation. For the present paper, pytorch provided the core neural network and training functionality.

3.10.1.0.2 HuggingFaces Huggingfaces packages provide for download, use and evaluation of transformer based models and benchmark datasets. The Huggingfaces website provides a repository for pretrained models, including commonly used models such as BERT.

3.10.1.0.3 Pennylane Pennylane[10] provides differentiable quantum programming and simulation, allowing integration of quantum circuits into general machine learning pipelines such as PyTorch[34].

[9] https://github.com/sikfeng/quantum-transfer-learning-with-bert/blob/main/Quantum_Transfer_Learning_with_BERT.ipynb

[10] a product of Xanadu. NB: Mari 2020 [5] was conducted by researchers at Xanadu

Chapter 4

Results

4.1 Training & Validation Performance

Training and validation performance for each circuit type is presented in figure 4.1, which aggregates results over all different qubit numbers. An equivalent set of graphs for accuracy is presented in 4.3.

We note first that the sharp drop in validation loss after 10 epochs is a consequence of the scheduler reducing the learning rate[1]. Other than the drop due to rate adaption, training and validation curves are closely aligned, suggesting the hybrid classifier is able to generalise to unknown data.

Undressed Amplitude Encoding circuit shows significant the worst performance, converging to a higher loss and lower accuracy for both training and validation. Undressed Amp. plateus at a minimum training loss of 0.652 and validation loss of 0.657. In general terms this behaviour is expected; the classifier of Undressed Amp. has fewer processing layers and far fewer trainable parameters than all other circuits.

All DQC show improved training performance to the purely classical model. While the classical model plateaus at 0.642 training loss, quantum models train to below 0.64. RY only and Encoding only reach 0.639, while RY+RZ and Dressed Amplitude encoding reaching 0.637 and 0.635 respectively. The improved training loss is not matched with reduced validation loss however, with all models stabilising at validation loss of 0.643.

It was expected that quantum circuits would allow for lower training loss, as the quantum circuit extends the expressive ability of the network. However, it is notable

[1]NB: using a lower learning rate by default does not result lower validation error from the start of training; the drop is a result of the reduction of rates, not of low rates

the circuit with no active rotation gates shows lower training loss, suggesting at least some of the improved performance is due to the normalisation and use of tanh rather than ReLU. I note BERT incorporates a tahn activation for the output layer[2]. The lower training loss for RZ+RY and Dressed Amp. may be due to greater power of the the quantum circuits, or due to the use of larger classical input layers.

Addressing training accuracy briefly, the quantum circuits show improved generalisation, having no gap between training and validation accuracy. This behavior is also shown by the inactive quantum circuit, but is not shown by the undressed quantum circuit. This suggests the behaviour is a consequence of a classical network learning to encode into a quantum circuit, rather than quantum processing itself.

Figure 4.3 considers training behaviour for circuits with differing numbers of qubits. As expected, wider circuits show lower loss after training. However, as this effect is also observed on purely classical circuits, it is unlikley to be an effect of greater quantum circuit expressivity.

4.2 Test Set Performance

We now consider the performance of the trained hybrid quantum classifier on the MRPC test set. A summary of test performance for each circuit is provided in tables 4.1 and 4.2. For conciseness, data is only provided for even qubit numbers; full results are provided in appendix B.

Considering first table 4.1 which compares the performance of circuits making use of angle encoding, we first note there are no consistent or significant differences between performance for the different ansatz. Additionally, to three significant figures there is no observable trend in loss with increase in qubit number. Loss remains within the range 0.655 ± 0.002 for every combination of circuit variant and qubit number, with differences in loss between circuits lying within the standard error in the mean. Accuracy and F1 vary more significantly but again show no consistent upward trend. Notably, quantum circuits show slightly improved test accuracy compared to the purely classical classifier.

Table 4.2 compares the performance of a circuit making use of Angle Encoding and RY gates, with a circuit making use of Amplitude Encoding and RY gates, dressed with a classical input layer or undressed. Again, no clear continuous trend of improvement with qubit number is observed. The dressed amplitude encoding circuit shows comparable

[2] https://docs.allennlp.org/v2.10.1/api/modules/seq2vec_encoders/bert_pooler/

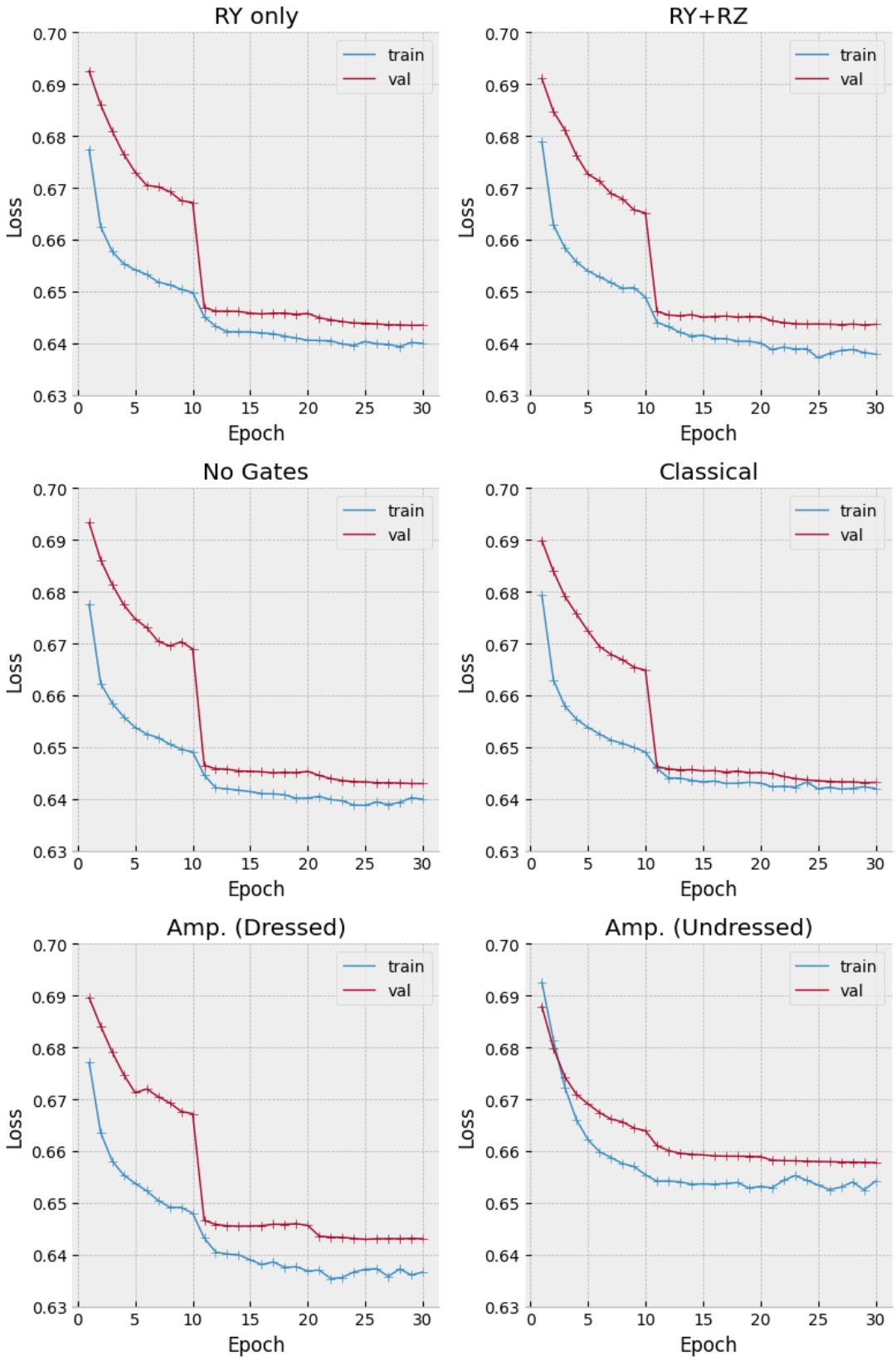

Figure 4.1: Figures showing mean training and validation loss for each circuit type, with results aggregated across all qubit numbers. Amp. (Dressed) comprises a classical input layer to map input features for amplitude encoding, while Amp. (Undressed) performs amplitude encoding directly.

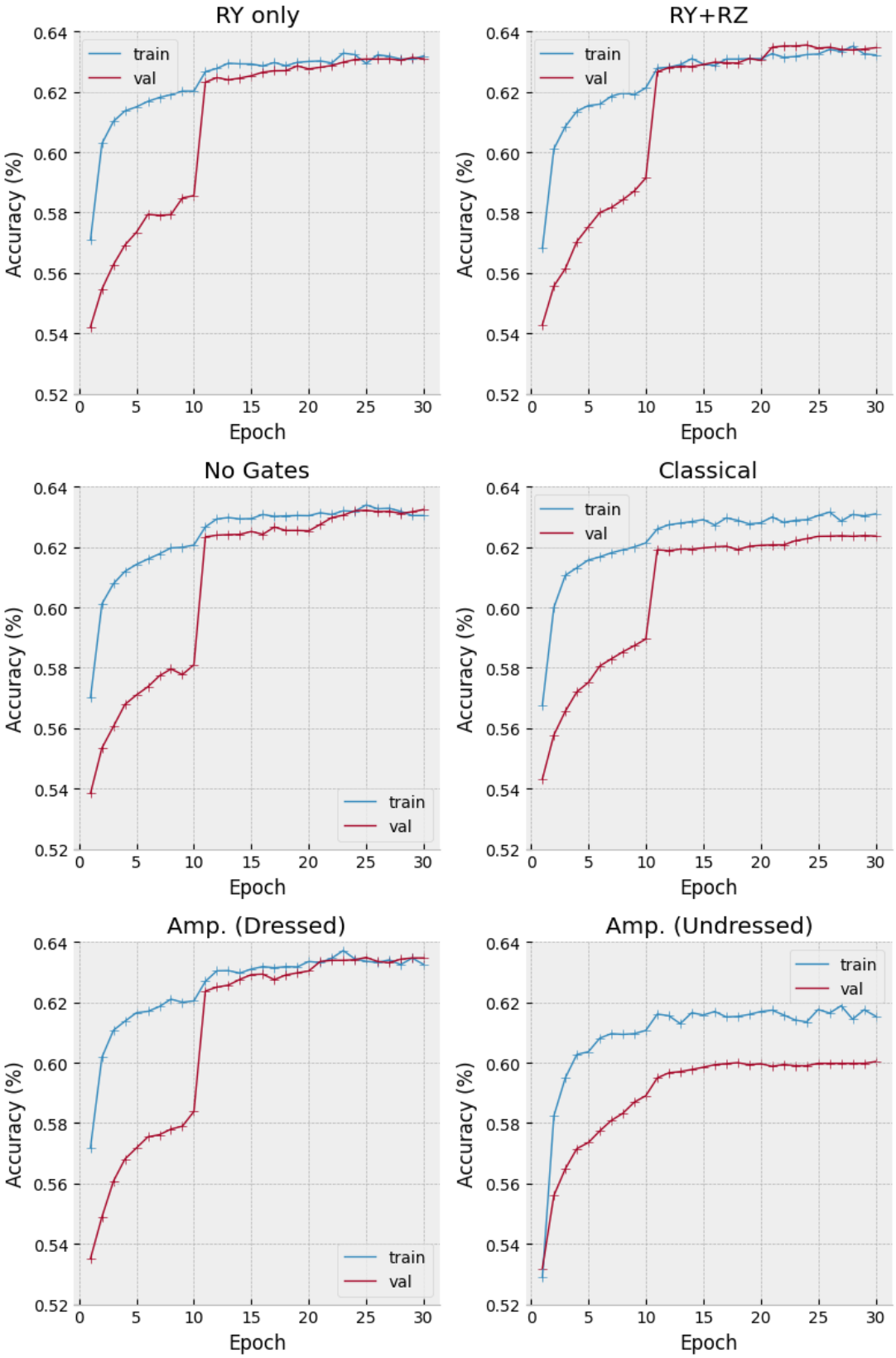

Figure 4.2: Figures showing mean training and validation accuracy for each circuit type, with results aggregated across all qubit numbers.

Type	Qubits	Loss	Accuracy (%)	F1	Time (m)
Classical	2	0.657 ± 0.002	0.618 ± 0.002	0.687 ± 0.001	1.7
No Gates	2	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.002	5
RY	2	0.655 ± 0.001	0.619 ± 0.002	0.688 ± 0.001	10
RY + RZ	2	0.656 ± 0.003	0.617 ± 0.003	0.683 ± 0.003	14
Classical	4	0.654 ± 0.001	0.618 ± 0.002	0.686 ± 0.001	1.6
No Gates	4	0.656 ± 0.001	0.62 ± 0.001	0.683 ± 0.002	5.5
RY	4	0.654 ± 0.0	0.621 ± 0.001	0.683 ± 0.001	15.8
RY + RZ	4	0.656 ± 0.001	0.619 ± 0.003	0.68 ± 0.003	23.4
Classical	6	0.656 ± 0.001	0.619 ± 0.001	0.685 ± 0.001	1.7
No Gates	6	0.654 ± 0.001	0.62 ± 0.001	0.684 ± 0.002	9.1
RY	6	0.654 ± 0.001	0.622 ± 0.001	0.682 ± 0.002	28.7
RY + RZ	6	0.657 ± 0.002	0.619 ± 0.003	0.674 ± 0.004	49.7
Classical	8	0.655 ± 0.001	0.617 ± 0.001	0.685 ± 0.001	1.6
No Gates	8	0.654 ± 0.001	0.622 ± 0.001	0.684 ± 0.001	11.9
RY	8	0.655 ± 0.001	0.62 ± 0.002	0.683 ± 0.002	43.4
RY + RZ	8	0.653 ± 0.001	0.618 ± 0.003	0.68 ± 0.001	71.9
Classical	10	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.001	1.6
No Gates	10	0.656 ± 0.001	0.619 ± 0.001	0.683 ± 0.002	16.1
RY	10	0.654 ± 0.001	0.624 ± 0.002	0.682 ± 0.002	58.6
RY + RZ	10	0.654 ± 0.001	0.622 ± 0.003	0.684 ± 0.002	119.6
Classical	12	0.655 ± 0.001	0.619 ± 0.002	0.685 ± 0.001	1.6
No Gates	12	0.655 ± 0.001	0.622 ± 0.001	0.684 ± 0.001	17.7
RY	12	0.656 ± 0.001	0.618 ± 0.002	0.678 ± 0.001	80.9
RY + RZ	12	0.654 ± 0.001	0.621 ± 0.002	0.674 ± 0.001	163.6
Classical	14	0.655 ± 0.001	0.617 ± 0.001	0.685 ± 0.001	1.6
No Gates	14	0.654 ± 0.001	0.618 ± 0.002	0.681 ± 0.002	25.9
RY	14	0.655 ± 0.001	0.619 ± 0.003	0.68 ± 0.002	118
RY + RZ	14	0.654 ± 0.001	0.62 ± 0.002	0.684 ± 0.002	179.9
Classical	16	0.655 ± 0.001	0.618 ± 0.002	0.683 ± 0.001	1.6
No Gates	16	0.654 ± 0.0	0.62 ± 0.002	0.682 ± 0.003	49.4
RY	16	0.654 ± 0.001	0.624 ± 0.001	0.681 ± 0.002	291.1
RY + RZ	16	0.656 ± 0.001	0.62 ± 0.003	0.679 ± 0.002	352.8
Classical	18	0.654 ± 0.001	0.618 ± 0.001	0.682 ± 0.001	1.8
No Gates	18	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.003	112.8
RY	18	0.655 ± 0.0	0.621 ± 0.002	0.68 ± 0.005	1122.9

Table 4.1: Table of test loss, accuracy, F1 and training times for classical, encoding only, RY only and RY+RZ circuits. Reported with standard error.

Type	Qubits	Loss	Accuracy (%)	f1
Amp Enc. (Dressed)	2	0.655 ± 0.001	0.623 ± 0.002	0.682 ± 0.002
RY	2	0.655 ± 0.001	0.619 ± 0.002	0.688 ± 0.001
Amp Enc. (Dressed)	4	0.654 ± 0.0	0.619 ± 0.002	0.68 ± 0.001
RY	4	0.654 ± 0.0	0.621 ± 0.001	0.683 ± 0.001
Amp Enc. (Dressed)	6	0.653 ± 0.0	0.624 ± 0.002	0.683 ± 0.001
RY	6	0.654 ± 0.001	0.622 ± 0.001	0.682 ± 0.002
Amp Enc. (Dressed)	8	0.653 ± 0.001	0.62 ± 0.002	0.682 ± 0.002
Amp Enc. (Undressed)	8	0.663 ± 0.002	0.596 ± 0.003	0.677 ± 0.002
RY	8	0.655 ± 0.001	0.62 ± 0.002	0.683 ± 0.002
Amp Enc. (Dressed)	10	0.654 ± 0.0	0.62 ± 0.001	0.678 ± 0.003
Amp Enc. (Undressed)	10	0.662 ± 0.001	0.601 ± 0.004	0.681 ± 0.002
RY	10	0.654 ± 0.001	0.624 ± 0.002	0.682 ± 0.002
Amp Enc. (Dressed)	12	0.652 ± 0.001	0.626 ± 0.004	0.68 ± 0.002
Amp Enc. (Undressed)	12	0.663 ± 0.0	0.602 ± 0.002	0.678 ± 0.002
RY	12	0.656 ± 0.001	0.618 ± 0.002	0.678 ± 0.001
Amp Enc. (Dressed)	14	0.654 ± 0.001	0.621 ± 0.002	0.679 ± 0.002
Amp Enc. (Undressed)	14	0.668 ± 0.002	0.598 ± 0.003	0.685 ± 0.002
RY	14	0.655 ± 0.001	0.619 ± 0.003	0.68 ± 0.002
Amp Enc. (Dressed)	16	0.653 ± 0.001	0.622 ± 0.002	0.681 ± 0.0
Amp Enc. (Undressed)	16	0.664 ± 0.002	0.597 ± 0.004	0.679 ± 0.003
RY	16	0.654 ± 0.001	0.624 ± 0.001	0.681 ± 0.002

Table 4.2: Table showing example test performance for circuits using amplitude encoding, with or without a classical input layer, compared to Angle Encoding with an input layer (referred to as RY). Undressed encoding is only possible for $\geq$ 7 qubits

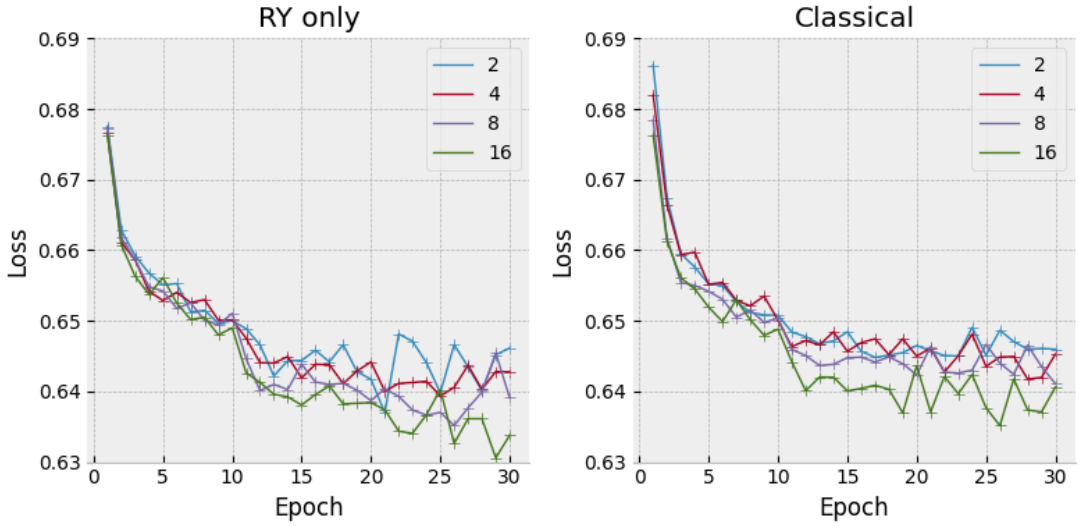

Figure 4.3: Figures showing training performance.

Circuit	Loss	Accuracy (%)	F1
RY	0.6547 ± 0.0006	0.6207 ± 0.0018	0.6819 ± 0.0020
RY+RZ	0.6550 ± 0.0011	0.6195 ± 0.0026	0.6798 ± 0.0022
No gates	0.6548 ± 0.0007	0.6195 ± 0.0015	0.6830 ± 0.0019
Amp. enc. (Dressed)	0.6536 ± 0.0005	0.6218 ± 0.0021	0.6806 ± 0.0017
Amp. enc. (Undressed)	0.6639 ± 0.0014	0.5988 ± 0.0031	0.6800 ± 0.0022
Classical	0.6550 ± 0.0009	0.6180 ± 0.0015	0.6843 ± 0.0011

Table 4.3: Loss, Accuracy and F1 for each circuit, aggregated across all circuit widths

performance to the dressed angle encoding circuit, showing equal or marginally lower loss for every tabulated qubit number and comparable prediction accuracy [3]. The undressed amp. encoding circuit shows significantly worse performance, with higher loss and lower accuracy for all qubit numbers.

Table 4.3 shows relevant metrics for each circuit when aggregating all runs, regardless of qubit number. It can be seen that, again, metrics for all circuits other than Undressed Amplitude Encoding are within standard error, and so are statistically indistinguishable. Dressed Amplitude Encoding shows lower loss and higher accuracy than other circuit designs, while Undressed Amplitude Encoding showing significantly worse performance. It is noted that Classical shows the highest F1 despite lower accuracy than all DQC, suggesting higher precision or recall.

[3]Training times are not reported, as amplitude encoding is accomplished using a simulator rather than apply sequential gates.

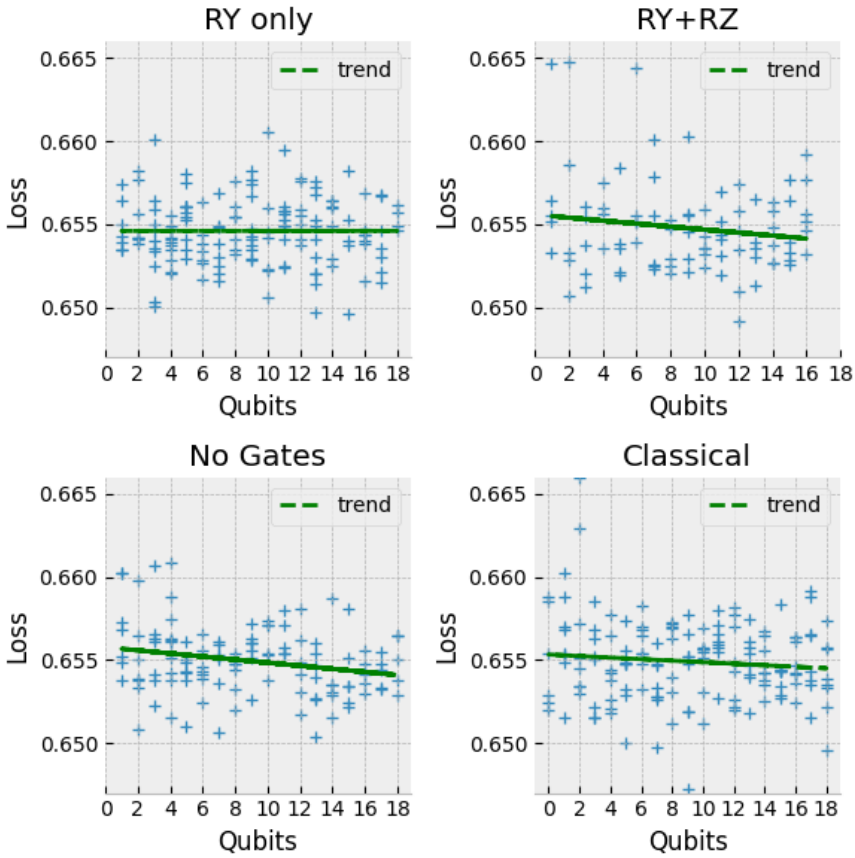

Figure 4.4: Graphs of Test Loss against Qubit number for Angle Encoding Circuits (RY ansatz, RY+RZ ansatz, No gates ansatz) and classical for comparison. Each graph displays a first order trendline. See appendix B.1 for full loss and accuracy graphs.

4.2.1 Loss & Accuracy trends

We now aim to determine how circuit performance scales with qubit number. Figure 4.4 shows test loss against qubit number for four circuits, with corresponding trendlines. See appendix B.1 for full graphs for all circuit types. For each circuit trend equations were determined by least squares fit for a first order polynomial.

We note first that the gradient for the RY circuit using angle encoding is positive, suggesting test performance degrades with qubit number. However, we also note the gradient is on the order of 10^{-7}, well below uncertainty of the data. To first order, the number of intermediate qubits are insignificant on the loss.

Other circuit architectures show more significant trends in loss with qubit number. Purely classical circuits show a slow reduction of loss with intermediate node numbers, possibly due to lower information loss during dimensionality reduction. Circuits making use of RY and RZ gates show higher rates of reduction; extrapolating the trendlines

Circuit Type	Loss Trend	Accuracy Trend
RY only	1.280e-07N_q+ 0.6546	2.552e-04N_q + 0.6177
RY + RZ	-8.995e-05N_q+ 0.6556	9.976e-05N_q + 0.6190
No Gates	-9.290e-05N_q+ 0.6558	-2.258e-05N_q + 0.6199
Classical	-4.610e-05N_q+ 0.6553	1.911e-04N_q + 0.6198
Amp. (Dressed)	-8.371e-05N_q+ 0.6546	-5.030e-04N_q + 0.6080
Amp. (Undressed)	2.707e-04N_q+ 0.6605	-1.631e-05N_q + 0.6178

Table 4.4: Extrapolated Loss and Accuracy trends for each circuit type.

predicts that RY+RZ circuits will outperform classical circuits for any width ≥ 6 qubits. Dressed Amplitude Encoding is predicted to have the lowest loss for all qubit numbers.

However, it is noted that the quantum circuit with no quantum processing shows the greatest rate of change with qubit number. With no obvious reason the ”no gates” circuit should scale faster than classical or RY only, it appears likely that all trends are dominated by noise.

Additionally, observing figure 4.4, one can see that the RY+RZ and Classical plots include several datapoints with losses significantly above the trendline. The observed downwards trend is likely caused by occasional training failures for low qubit numbers.

Considering instead trends in prediction accuracy; classical, RY and RY+RZ circuits show increasing accuracy for wider circuits, while other circuits were determined to show decreasing performance. Only RY was determined to scale faster than Classical. If linear trends continue, RY only quantum circuits will outperform classical circuits for greater than 33 qubit circuits.

4.3 Reassessment of Mari 2020

As set out in 3.7, the results set out above suggest there is limited, if any, benefit from the use of QTL for MRPC. Of particular note, many features believed to result from the use a quantum processor are replicated by a circuit with no active gates.

However, it is possible this limited performance is due to specifics of the present problem, such as the limitations of tinyBERT or the difficulty of MRPC. To this end, we briefly evaluate the identified circuits on the ant/bee classification of Mari [5]

Note that Mari does not provide a test set, instead reporting the best performance on the validation set. As such, reported performance will likely be better than actual

performance on unknown data. Additionally, as only small scale experiments are being conducted, we cannot determine uncertainty in the results.

When averaged over all trialed circuit, we observe that, again, the undressed circuit shows significantly the worst performance for both loss and accuracy. Both angle encoding circuits (RY or RY+RZ) show marginally improved accuracy over a classical output layer. RY+RZ shows both improved loss and accuracy. Notably, Dressed Amplitude Embedding shows high loss but also high accuracy, suggesting either low scores for accurate classes or occasional confident wrong predictions.

Type	Loss	Accuracy (%)
Classical	0.1822	0.9635
Embedding Only	0.1569	0.9822
Amp Emb. (Undressed)	0.6776	0.5552
RY only	0.1913	0.9757
RY+RY	0.1742	0.9795
Amp. Emb (Dressed)	0.2838	0.9841

Table 4.5: Performance for each type of circuit on ants/bees classification using transfer learning with ResNet18, when aggregated over all qubit numbers

However, the best performing option by loss, and second best for accuracy, is Embedding Only, the circuit with no active quantum gates. This suggests that the perceived benefits of the DQC are not due to processing carried out by the VQA.

We note finally that while operation with tinyBERT showed very small changes of loss with qubit number, Mari shows more significant effects:

- RY only: Loss $\approx -0.01022602\mathrm{N}_q + 0.34484898$
- RY + RZ: Loss $\approx -0.00494201\mathrm{N}_q + 0.26479966$
- Classical: Loss $\approx -0.00465374\mathrm{N}_q + 0.23161293$

By which the RY only method will out perform a classical circuit for any circuit wider than 18 qubits, while RY+RZ requires 110.

4.4 Reassessment with Finetuned BERTbase

The above data suggests minimal functional differences between quantum transfer learning and training a classical classification head, regardless of circuit width. However, it is possible the stagnant performance is due to a poor choice of classical model; the Dressed Quantum circuits may be limited by the low quality outputs of tinyBERT.

Type	Loss	Accuracy (%)	f1
Amp Enc. (Dressed)	0.548853	0.790247	0.808576
Amp Enc. (Undressed)	0.655343	0.771114	0.783729
Classical	0.691958	0.782032	0.806784
Encoding Only	0.587022	0.786950	0.809294
RY	0.602461	0.789411	0.808872
RY + RZ	0.578994	0.787253	0.808259

Table 4.7: Summary of performance of Quantum Transfer Learning on MRPC, with the classical input network being BERTbase finetuned on MRPC.

To determine if this is the case, previous experiments were rerun, with BERTbase (12 layers of 756 nodes, with 12 attention heads) finetuned on MRPC [4], trained for 15 epochs rather than 30. For conciseness, relevant data is included in the appendix, and briefly summarised in table 4.6.

Type	Qubits	Loss	Acc
Amp Enc. (Dressed)	4	0.3895	0.9505
Classical	4	0.1999	0.9584
RY	4	0.3183	0.9307
RY + RZ	4	0.2579	0.9464
Amp Enc. (Dressed)	8	0.3244	0.9732
Classical	8	0.2010	0.9534
RY	8	0.2696	0.9383
RY + RZ	8	0.2171	0.9542
Amp Enc. (Dressed)	12	0.2691	0.9693
Amp Enc. (Undressed)	12	0.6816	0.5348
Classical	12	0.1583	0.9689
RY	12	0.2386	0.9367
RY + RZ	12	0.1988	0.9556
Amp Enc. (Dressed)	16	0.4360	0.9602
Amp Enc. (Undressed)	16	0.6908	0.5000
Classical	16	0.1599	0.9686
RY	16	0.1826	0.9543
RY + RZ	16	0.1922	0.9529

Table 4.6: Performance on ants/bees classification using transfer learning with ResNet18, by qubit numbers and circuit

Notably, Dressed Amplitude encoding, No Gates and RY+RZ show the lowest loss and highest accuracy. It should be noted that dressed amplitude encoding and RY+RZ also contain large input classical circuits, but the performance of No Gates is more notable. We also note, Undressed Amplitude encoding shows lower loss than Classical, but also lower accuracy. The high loss but high accuracy of classical may be due to occasional highly confidently wrong predictions. The performance of 79% accuracy, 81% F1 is worse than the models reported performance (86/90 acc./F1), which is to be expected when using a smaller normalised set of MRPC data.

[4] provided by `https://huggingface.co/Intel/bert-base-uncased-mrpc`

Chapter 5

Discussion

In the preceding section we have demonstrated the utility of Quantum Transfer Learning with a reduced size model of BERT, finding performance comparable with simple classical models. We have implemented novel variations of Dressed Quantum Circuits, including different ansatz and embeddings (see 4.3). In particular, we have aimed to determine the impact of the classical dressing layers.

Firstly, we observed that all variations of the Dressed Quantum Circuit (DQC) showed similar test performance after training on MRPC. All minor variations fell within the standard error of the results. In addition, we aimed to determine the impact of changes to circuit width (qubit number) to ascertain if quantum advantage is readily realisable or requires significant improvement in the scale of quantum devices. Unfortunately, for the classical network and problem chosen, loss and accuracy gradients were found to be small or positive (implying worsening performance with quantum size), dominated by the randomness of the training process.

Stronger effects were observed during the training process. Dressed circuits showed faster training than comparable classical networks. In particular, two circuits showed notably low training loss: an ansatz combining Pauli Y and Pauli Z rotations with angle embedding; and a ansatz making use of only Pauli Y rotations but utilising amplitude embedding to encode far more features. These two circuits showed consistent, if minor, improvements over classical models and previously used RY only DQC. We do note however that these two circuits made use of the largest classical input layers, and so performance may not be due to quantum expressive power.

However, many of the seemingly advantageous features of QTL over classical output circuit, such as the slight reduction in training loss and slight increase in training accuracy are replicated by quantum circuits with no active rotation gates, as described in

3.4.4. These circuits, referred to herein as ”No Gates” or ”Encoding Only” are variants of DQC that make use of angle encoding to embed input features then immediately measure the outputs. The performance of these circuits is comparable to other assessed ansatz, despite the quantum circuit only acting as an unconventional activation function for a classical network.

Moreover, we have investigated the performance of undressed Variational Quantum Circuits (VQC), making use of amplitude encoding to avoid the need for an input classical layer. Said circuits show consistently low performance, during both training and testing. While amp. encoding allow for exponentially more features to be encoded into the circuit, this encoding is not trained for a specific purpose. Classical encoders, in contrast, are able to reduce dimensionality while also performing useful processing.

The high performance of a DQC without a quantum core and the low performance of a quantum circuit without dressing, taken together, strongly suggest that the classical layers of the DQC are not acting as means to connect to a powerful VQC. Instead, the classical layers are performing the relevant processing themselves, with the quantum circuit applying an activation that is convenient for classification. Notably, the circuit applies a hyperbolic tangent function, which is commonly used in classifiers due to centering inputs and providing for positive or negative values, unlike ReLU.

To explore the possibility that the problem chosen was simply inappropriate for QTL, we have applied the same techniques to re-evaluate the classifiers of Mari 2020. We observed lower loss and higher accuracy for classification using ResNet18, but equivalent results in terms of which circuits gave the best accuracy and loss.

While we have not observed a clear quantum advantage arising from QTL, or determined a circuit width likely to show such), the method of DQC allows for simple connection of arbitrary quantum circuit to arbitrary classical inputs. The techniques here can be readily applied to deep quantum networks with alternative ansatz or methods of encoding.

5.1 Limitations and Future Work

There remains open the possibility that none of the circuits considered allow for meaningful quantum computation. It must be noted that the preceding experiments did not consider the effect of changing quantum circuit depth, using a circuit of six layers for all experiments. This depth was selected to allow easy comparison to previous work in the field (see [5, 13] for example), and to avoid multiplying the number of experiments that

needed to be undertaken. However, it is likely that the benefits of variational quantum circuits emerge from deep layered circuits [2, 19].

Additionally, while some hyperparameters (learning rate, batch size, epochs) were chosen to give the best training performance on classical and quantum using y gates, no additional tuning. This could be conducted in the future. In particular, the quantum circuits were initialised with small Gaussian values (0.01), which likely results in the initial quantum circuit having very weak effects; the trace distance between $RY(0.01)$ and the identity is 0.005. As such it is possible the quantum network initialises in portion of loss space in which the quantum weights have low contribution and the classical weights large contributions, and subsequently learns to keep quantum weights low. Briefly, the quantum parameters of a random sample of trained models were viewed, finding post training angles remaining on the order of 0.01 or less. Using a large initial spread of quantum values may allow for better investigation of the impact of the quantum circuit.

5.2 Encoding Only Circuit Benchmark

Finally, we propose the use of an encoding only Dressed Quantum Circuit as a benchmark for future work in quantum transfer learning. As noted above, despite lacking the active elements of a VQA, it shows similar performance. Use as a benchmark removes the impact of, for example, different normalisations. As the frozen quantum circuit can be represented as as simple equation (see D for details), future work could investigate replacing the frozen quantum circuit with a modified activation function.

5.2.1 Relevant Literature

Re-evaluating the most immediately relevant work, Buonaiuto et al. 2024, [13] have previously considered both amplitude encoding and quantum transfer learning with BERT, training for Italian CoLA. They observed comparable performance to purely classical output layers. This is a notable contradiction to the present work, which found significantly worse performance with the undressed amplitude circuits. Buonaito uses six layers of conventional single rotation + CNOT ansatz. It is possible Buonaito determined functioning hyperparmeters allowing for functional training, that the technique makes use of an unstated input layer, or that the problem is better suited to the technique.

Buonaito performs a qualitative analysis of the performance of their hybrid quantum

classical classifier, noting that the encoding provides benefits for problems requiring longer more structured inputs but worse performance for short and simple inputs, which they ascribe to the nature of amplitude encoding; while it allows for exponential encoding, it also requires an exponential number of queries to extract said information [35]. They argue this reduces the importance of each individual token but allows for investigation of structure. We note that MRPC, relating to paraphrase identification, contains mostly short inputs and relies on the meaning of individual words.

Chapter 6

Conclusion

In the preceding section we have demonstrated the utility of Quantum Transfer Learning with a reduced size model of BERT, finding performance comparable with simple classical models. We have implemented novel variations of Dressed Quantum Circuits, including different ansatz and embeddings. When evaluated on MRPC all variations and qubit numbers showed indistinguishable performance, suggesting that, for the selected problem at least, the quantum processing has minimal effect.

Quantum circuits were observed to provide minor training improvements over purely classical outputs, with better training loss for more complex ansatz and larger qubit number. However, this performance did not generalise to validation or test sets. The more complex ansatz considered, making use of two rotations about different axes rather than one, showed marginally reduced minimum training loss but no improvement on the test set. We also note that the more complex ansatz included larger classical input layers, and so it is difficult to ascribe credit for the performance.

It is possible these limited effects are due to the nature of the BERT classifier, intended for all processing to be performed in the deep network, with the output intended to be processed by a single classical output layer. It is also possible the problem selected was too simple to benefit from the expressively of quantum circuits. Furthermore, it is possible the initialisation is not efficient for the chosen structure.

To investigate these effects we reassessed the work of Mari 2020 [5], which originally motivated the use of Dressed Quantum Circuits, similarly finding no performance improvement with use of quantum networks. For the reassessment of Mari, however, there was a clearly observable trend in improvement with larger qubit numbers, suggesting future large circuits may provide a benefit.

Notably, we observed that several ”benefits of quantum circuits” are realised even

with no active quantum gates. This suggests the Dressed Quantum Circuit is not acting a quantum processor, but as a small classical network. We suggest said Encoding Only Circuits can be used as a benchmark for future work in the field.

Similarly, we made use of amplitude encoding to remove one of the classical layers, observing significantly reduced performance. This, combined with the performance of the circuits with no active gates again hints that the much of the processing power of a DQC originates from the classical input and output layers.

Finally, we repeat that we have not provided an improved classifier, not have we sought to. Experimentation has aimed only to investigate potential improvements and benchmarking methods for the quantum components of Quantum Transfer Learning.

Overall, we have not been able to observe quantum supremacy or a likely path using Quantum Transfer Learning. However, dressed quantum circuits did prove to be flexible and simple means for connecting classical and quantum systems. The present work has relied on the ability to flexibly change quantum circuits without adapting the classical or vice versa.

6.0.1 Future Work

We recommend that future work investigate deeper quantum circuits. The shallow circuits in use presently do not appear to offer significant quantum advantages when used with quantum transfer learning. We also suggest that future work investigates hyperparameter tuning for quantum circuit parameters, in particular considering methods of initialisation.

As well as deeper circuits, future work could aim to quantise the entire pipeline, or refine the network specifically for quantum outputs. For example [36] proposes a fully quantum transformer architecture. [37] develops on previous discussions, proposing novel methods of deep training BERT models to be more appropriate for use with a quantum classifier head, such as requiring normalisation at all hidden layers, and adjusting next sentence prediction to replicate variational measurement. The authors observed overall performance comparable to classical BERT, with significant throughput increase for quantum models.

Chapter 7

References

[1] Kai Li et al. “Quantum Algorithms for Solving Linear Regression Equation”. en. In: *Journal of Physics: Conference Series* 1738.1 (Jan. 2021). Publisher: IOP Publishing, p. 012063. ISSN: 1742-6596. DOI: 10.1088/1742-6596/1738/1/012063. URL: https://dx.doi.org/10.1088/1742-6596/1738/1/012063.

[2] Amira Abbas et al. “The power of quantum neural networks”. en. In: *Nature Computational Science* 1.6 (June 2021). Publisher: Nature Publishing Group, pp. 403–409. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00084-1. URL: https://www.nature.com/articles/s43588-021-00084-1.

[3] Maria Schuld et al. “Circuit-centric quantum classifiers”. In: *Phys. Rev. A* 101 (3 Mar. 2020), p. 032308. DOI: 10.1103/PhysRevA.101.032308. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.032308.

[4] Andrea Mari. “Quantum transfer learning”. en. In: *PennyLane Demos* (Dec. 2019). Publisher: Xanadu. URL: https://pennylane.ai/qml/demos/tutorial_quantum_transfer_learning/.

[5] Andrea Mari et al. “Transfer learning in hybrid classical-quantum neural networks”. en-GB. In: *Quantum* 4 (Oct. 2020). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 340. DOI: 10.22331/q-2020-10-09-340. URL: https://quantum-journal.org/papers/q-2020-10-09-340/.

[6] Kanimozhi T et al. “Brain Tumor Recognition based on Classical to Quantum Transfer Learning”. In: *2022 International Conference on Innovative Trends in Information Technology (ICITIIT)*. Feb. 2022, pp. 1–5. DOI: 10.1109/ICITIIT5

4346.2022.9744220. URL: https://ieeexplore.ieee.org/abstract/document/9744220.

[7] Vanda Azevedo, Carla Silva, and Inês Dutra. "Quantum transfer learning for breast cancer detection". en. In: *Quantum Machine Intelligence* 4.1 (Feb. 2022), p. 5. ISSN: 2524-4914. DOI: 10.1007/s42484-022-00062-4. URL: https://doi.org/10.1007/s42484-022-00062-4.

[8] Erdi Acar and İhsan Yilmaz. "COVID-19 detection on IBM quantum computer with classical-quantum transfer learning: Turkish Journal of Electrical Engineering & Computer Sciences". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 29.1 (Jan. 2021). Publisher: Scientific and Technical Research Council of Turkey, pp. 46–61. ISSN: 13000632. DOI: 10.3906/elk-2006-94. URL: https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=148488083&site=ehost-live.

[9] Muhammad Junaid Umer et al. "An integrated framework for COVID-19 classification based on classical and quantum transfer learning from a chest radiograph". English. In: *Concurrency and Computation: Practice and Experience* (June 2021). Place: Hoboken, United States Publisher: John Wiley & Sons, Inc. DOI: 10.1002/cpe.6434. URL: https://www.proquest.com/docview/2546496657/abstract/BD503FC5BBEA49B4PQ/1.

[10] Sridevi S et al. "Quantum Transfer Learning for Diagnosis of Diabetic Retinopathy". In: *2022 International Conference on Innovative Trends in Information Technology (ICITIIT)*. Feb. 2022, pp. 1–5. DOI: 10.1109/ICITIIT54346.2022.9744184. URL: https://ieeexplore.ieee.org/abstract/document/9744184.

[11] Toshiaki Koike-Akino, Pu Wang, and Ye Wang. "Quantum Transfer Learning for Wi-Fi Sensing". In: *ICC 2022 - IEEE International Conference on Communications*. ISSN: 1938-1883. May 2022, pp. 654–659. DOI: 10.1109/ICC45855.2022.9839011. URL: https://ieeexplore.ieee.org/abstract/document/9839011.

[12] Bidisha Dhara, Monika Agrawal, and Sumantra Dutta Roy. "Multi-class classification using quantum transfer learning". en. In: *Quantum Information Processing* 23.2 (Jan. 2024), p. 34. ISSN: 1573-1332. DOI: 10.1007/s11128-023-04237-1. URL: https://doi.org/10.1007/s11128-023-04237-1.

[13] Giuseppe Buonaiuto et al. “Quantum transfer learning for acceptability judgements”. en. In: *Quantum Machine Intelligence* 6.1 (Mar. 2024), p. 13. ISSN: 2524-4914. DOI: 10.1007/s42484-024-00141-8. URL: https://doi.org/10.1007/s42484-024-00141-8.

[14] Ebrahim Ardeshir-Larijani and Mohammad Mahdi Nasiri Fatmehsari. “Hybrid classical-quantum transfer learning for text classification”. en. In: *Quantum Machine Intelligence* 6.1 (Mar. 2024), p. 19. ISSN: 2524-4914. DOI: 10.1007/s42484-024-00147-2. URL: https://doi.org/10.1007/s42484-024-00147-2.

[15] Junhan Qin. *Review of Ansatz Designing Techniques for Variational Quantum Algorithms*. 2022. arXiv: 2212.04913 [quant-ph]. URL: https://arxiv.org/abs/2212.04913.

[16] Abhinav Kandala et al. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. en. In: *Nature* 549.7671 (Sept. 2017). Publisher: Nature Publishing Group, pp. 242–246. ISSN: 1476-4687. DOI: 10.1038/nature23879. URL: https://www.nature.com/articles/nature23879.

[17] Yuxuan Du et al. “Quantum circuit architecture search for variational quantum algorithms”. en. In: *npj Quantum Information* 8.1 (May 2022). Publisher: Nature Publishing Group, pp. 1–8. ISSN: 2056-6387. DOI: 10.1038/s41534-022-00570-y. URL: https://www.nature.com/articles/s41534-022-00570-y.

[18] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. “Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms”. In: *Advanced Quantum Technologies* 2.12 (Oct. 2019). ISSN: 2511-9044. DOI: 10.1002/qute.201900070. URL: http://dx.doi.org/10.1002/qute.201900070.

[19] Edward Farhi and Hartmut Neven. *Classification with Quantum Neural Networks on Near Term Processors*. 2018. arXiv: 1802.06002 [quant-ph]. URL: https://arxiv.org/abs/1802.06002.

[20] Ashish Vaswani et al. *Attention Is All You Need*. en. June 2017. URL: https://arxiv.org/abs/1706.03762v7.

[21] Iulia Turc et al. *Well-Read Students Learn Better: On the Importance of Pre-training Compact Models*. arXiv:1908.08962 [cs]. Sept. 2019. DOI: 10.48550/arXiv.1908.08962. URL: http://arxiv.org/abs/1908.08962.

[22] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[23] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[24] Dongseong Hwang. *FAdam: Adam is a natural gradient optimizer using diagonal empirical Fisher information*. July 2024. DOI: 10.48550/arXiv.2405.12807. URL: http://arxiv.org/abs/2405.12807.

[25] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen, Grzegorz Chrupała, and Afra Alishahi. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: https://aclanthology.org/W18-5446.

[26] Alex Wang et al. *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. 2020. arXiv: 1905.00537 [cs.CL]. URL: https://arxiv.org/abs/1905.00537.

[27] William B. Dolan and Chris Brockett. "Automatically Constructing a Corpus of Sentential Paraphrases". In: *Proceedings of the Third International Workshop on Paraphrasing ({IWP}2005)*. 2005. URL: https://aclanthology.org/I05-5002.

[28] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. *Neural Network Acceptability Judgments*. Oct. 2019. DOI: 10.48550/arXiv.1805.12471. URL: http://arxiv.org/abs/1805.12471.

[29] Richard Socher et al. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Ed. by David Yarowsky et al. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. URL: https://aclanthology.org/D13-1170.

[30] Harshit Mogalapalli et al. “Trash classification using quantum transfer learning”. In: *AIP Conference Proceedings* 2424.1 (Mar. 2022), p. 070003. ISSN: 0094-243X. DOI: 10.1063/5.0076837. URL: https://doi.org/10.1063/5.0076837.

[31] Manuela Weigold et al. “Encoding patterns for quantum algorithms”. en. In: *IET Quantum Communication* 2.4 (2021), pp. 141–152. ISSN: 2632-8925. DOI: 10.1049/qtc2.12032. URL: https://onlinelibrary.wiley.com/doi/abs/10.1049/qtc2.12032.

[32] Mikko Mottonen et al. *Transformation of quantum states using uniformly controlled rotations*. 2004. arXiv: quant-ph/0407010 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0407010.

[33] Martin Plesch and Časlav Brukner. “Quantum-state preparation with universal gate decompositions”. In: *Phys. Rev. A* 83 (3 Mar. 2011), p. 032302. DOI: 10.1103/PhysRevA.83.032302. URL: https://link.aps.org/doi/10.1103/PhysRevA.83.032302.

[34] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2022. arXiv: 1811.04968 [quant-ph]. URL: https://arxiv.org/abs/1811.04968.

[35] Nathan Wiebe. “Key questions for the quantum machine learner to ask themselves”. In: *New Journal of Physics* 22.9 (Sept. 2020), p. 091001. DOI: 10.1088/1367-2630/abac39. URL: https://dx.doi.org/10.1088/1367-2630/abac39.

[36] Guangxi Li, Xuanqiang Zhao, and Xin Wang. “Quantum self-attention neural networks for text classification”. en. In: *Science China Information Sciences* 67.4 (Mar. 2024), p. 142501. ISSN: 1869-1919. DOI: 10.1007/s11432-023-3879-7. URL: https://doi.org/10.1007/s11432-023-3879-7.

[37] Qiuchi Li et al. *Adapting Pre-trained Language Models for Quantum Natural Language Processing*. Feb. 2023. DOI: 10.48550/arXiv.2302.13812. URL: http://arxiv.org/abs/2302.13812.

Appendix A

Additional figures

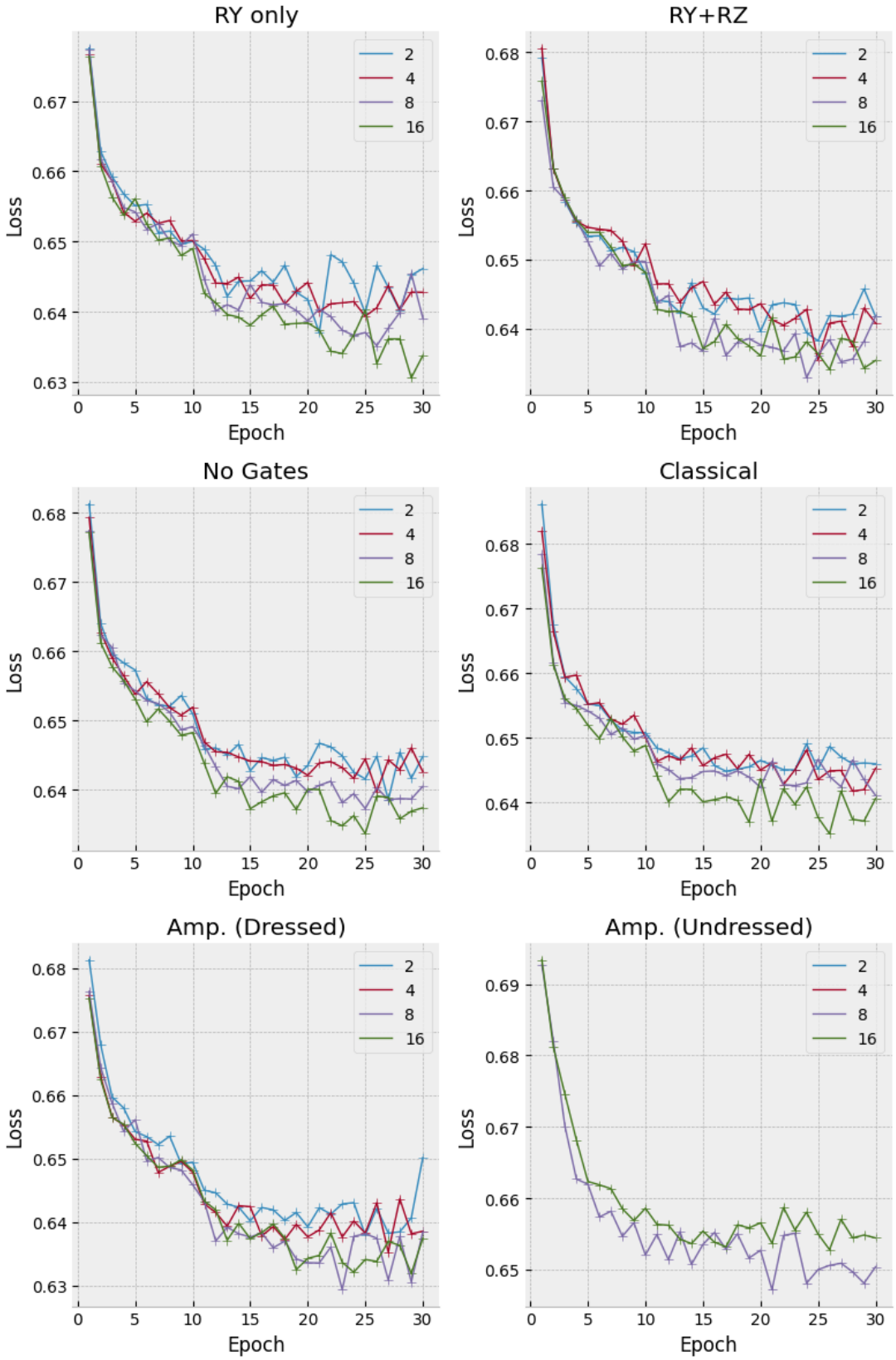

Figure A.1: Figures showing training performance for each circuit for selected qubit numbers

Appendix B

Full Test Data

The below table includes data for each circuit evaluated on the test set for each qubit number. Classical zero qubits indicates no intermediate layer.

Type	Qubits	Loss	Accuracy (%)	f1	Time (m)
Classical	0	0.655 ± 0.001	0.619 ± 0.001	0.682 ± 0.002	2.1
Amp Enc. (Dressed)	1	0.656 ± 0.001	0.614 ± 0.003	0.681 ± 0.002	6.9
Classical	1	0.656 ± 0.001	0.616 ± 0.002	0.682 ± 0.002	1.6
No Gates	1	0.657 ± 0.001	0.619 ± 0.001	0.68 ± 0.003	3.6
RY	1	0.655 ± 0.001	0.618 ± 0.001	0.685 ± 0.002	6.5
RY + RZ	1	0.657 ± 0.002	0.616 ± 0.002	0.68 ± 0.008	8.1
Amp Enc. (Dressed)	2	0.655 ± 0.001	0.623 ± 0.002	0.682 ± 0.002	15.3
Classical	2	0.657 ± 0.002	0.618 ± 0.002	0.687 ± 0.001	1.7
No Gates	2	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.002	5
RY	2	0.655 ± 0.001	0.619 ± 0.002	0.688 ± 0.001	10
RY + RZ	2	0.656 ± 0.003	0.617 ± 0.003	0.683 ± 0.003	14
Amp Enc. (Dressed)	3	0.655 ± 0.001	0.622 ± 0.001	0.684 ± 0.003	16.4
Classical	3	0.655 ± 0.001	0.616 ± 0.002	0.684 ± 0.001	1.6
No Gates	3	0.656 ± 0.001	0.616 ± 0.003	0.685 ± 0.001	6.1
RY	3	0.654 ± 0.001	0.617 ± 0.001	0.683 ± 0.002	12.7
RY + RZ	3	0.654 ± 0.001	0.623 ± 0.003	0.685 ± 0.001	22.8
Amp Enc. (Dressed)	4	0.654 ± 0.0	0.619 ± 0.002	0.68 ± 0.001	17.4
Classical	4	0.654 ± 0.001	0.618 ± 0.002	0.686 ± 0.001	1.6
No Gates	4	0.656 ± 0.001	0.62 ± 0.001	0.683 ± 0.002	5.5
RY	4	0.654 ± 0.0	0.621 ± 0.001	0.683 ± 0.001	15.8
RY + RZ	4	0.656 ± 0.001	0.619 ± 0.003	0.68 ± 0.003	23.4

Amp Enc. (Dressed)	5	0.654 ± 0.001	0.62 ± 0.002	0.683 ± 0.001	22.6
Classical	5	0.654 ± 0.001	0.617 ± 0.001	0.684 ± 0.002	1.8
No Gates	5	0.654 ± 0.001	0.625 ± 0.001	0.685 ± 0.002	8.2
RY	5	0.656 ± 0.0	0.618 ± 0.003	0.687 ± 0.002	20.3
RY + RZ	5	0.654 ± 0.001	0.62 ± 0.003	0.679 ± 0.003	38.6
Amp Enc. (Dressed)	6	0.653 ± 0.0	0.624 ± 0.002	0.683 ± 0.001	28
Classical	6	0.656 ± 0.001	0.619 ± 0.001	0.685 ± 0.001	1.7
No Gates	6	0.654 ± 0.001	0.62 ± 0.001	0.684 ± 0.002	9.1
RY	6	0.654 ± 0.001	0.622 ± 0.001	0.682 ± 0.002	28.7
RY + RZ	6	0.657 ± 0.002	0.619 ± 0.003	0.674 ± 0.004	49.7
Amp Enc. (Dressed)	7	0.652 ± 0.001	0.626 ± 0.001	0.679 ± 0.001	33.5
Amp Enc. (Undressed)	7	0.662 ± 0.002	0.607 ± 0.002	0.683 ± 0.002	37.5
Classical	7	0.653 ± 0.001	0.621 ± 0.001	0.683 ± 0.001	1.6
No Gates	7	0.655 ± 0.001	0.621 ± 0.002	0.682 ± 0.001	9.8
RY	7	0.654 ± 0.001	0.62 ± 0.001	0.684 ± 0.001	36.7
RY + RZ	7	0.655 ± 0.001	0.622 ± 0.002	0.685 ± 0.003	69.8
Amp Enc. (Dressed)	8	0.653 ± 0.001	0.62 ± 0.002	0.682 ± 0.002	42
Amp Enc. (Undressed)	8	0.663 ± 0.002	0.596 ± 0.003	0.677 ± 0.002	56.5
Classical	8	0.655 ± 0.001	0.617 ± 0.001	0.685 ± 0.001	1.6
No Gates	8	0.654 ± 0.0	0.622 ± 0.001	0.684 ± 0.001	11.9
RY	8	0.655 ± 0.001	0.62 ± 0.002	0.683 ± 0.002	43.4
RY + RZ	8	0.653 ± 0.001	0.618 ± 0.003	0.68 ± 0.001	71.9
Amp Enc. (Dressed)	9	0.654 ± 0.001	0.623 ± 0.002	0.681 ± 0.001	48.7
Amp Enc. (Undressed)	9	0.663 ± 0.0	0.607 ± 0.002	0.683 ± 0.002	57.6
Classical	9	0.654 ± 0.001	0.619 ± 0.001	0.684 ± 0.002	1.8
No Gates	9	0.656 ± 0.001	0.62 ± 0.001	0.684 ± 0.002	13
RY	9	0.655 ± 0.001	0.618 ± 0.002	0.681 ± 0.001	43.5
RY + RZ	9	0.655 ± 0.001	0.619 ± 0.003	0.679 ± 0.001	100.4
Amp Enc. (Dressed)	10	0.654 ± 0.0	0.62 ± 0.001	0.678 ± 0.003	58.5
Amp Enc. (Undressed)	10	0.662 ± 0.001	0.601 ± 0.004	0.681 ± 0.002	74.7
Classical	10	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.001	1.6
No Gates	10	0.656 ± 0.001	0.619 ± 0.001	0.683 ± 0.002	16.1
RY	10	0.654 ± 0.001	0.624 ± 0.002	0.682 ± 0.002	58.6
RY + RZ	10	0.654 ± 0.0	0.622 ± 0.003	0.684 ± 0.002	119.6
Amp Enc. (Dressed)	11	0.654 ± 0.001	0.62 ± 0.001	0.681 ± 0.001	71.9

Amp Enc. (Undressed)	11	0.665 ± 0.001	0.604 ± 0.004	0.681 ± 0.004	100.5
Classical	11	0.656 ± 0.001	0.617 ± 0.002	0.682 ± 0.002	1.8
No Gates	11	0.656 ± 0.001	0.616 ± 0.002	0.677 ± 0.002	16.3
RY	11	0.655 ± 0.001	0.616 ± 0.003	0.68 ± 0.002	57.1
RY + RZ	11	0.654 ± 0.001	0.621 ± 0.003	0.678 ± 0.004	133.8
Amp Enc. (Dressed)	12	0.652 ± 0.001	0.626 ± 0.004	0.68 ± 0.002	85
Amp Enc. (Undressed)	12	0.663 ± 0.0	0.602 ± 0.002	0.678 ± 0.002	115.8
Classical	12	0.655 ± 0.001	0.619 ± 0.002	0.685 ± 0.001	1.6
No Gates	12	0.655 ± 0.001	0.622 ± 0.001	0.684 ± 0.001	17.7
RY	12	0.656 ± 0.001	0.618 ± 0.002	0.678 ± 0.001	80.9
RY + RZ	12	0.654 ± 0.001	0.621 ± 0.002	0.674 ± 0.001	163.6
Amp Enc. (Dressed)	13	0.653 ± 0.001	0.627 ± 0.001	0.684 ± 0.001	136.1
Amp Enc. (Undressed)	13	0.662 ± 0.001	0.607 ± 0.004	0.683 ± 0.002	123.1
Classical	13	0.655 ± 0.001	0.614 ± 0.001	0.682 ± 0.001	1.8
No Gates	13	0.654 ± 0.001	0.618 ± 0.002	0.681 ± 0.001	16.8
RY	13	0.654 ± 0.001	0.623 ± 0.002	0.682 ± 0.001	76.1
RY + RZ	13	0.654 ± 0.001	0.621 ± 0.002	0.677 ± 0.003	199.8
Amp Enc. (Dressed)	14	0.654 ± 0.001	0.621 ± 0.002	0.679 ± 0.002	156
Amp Enc. (Undressed)	14	0.668 ± 0.002	0.598 ± 0.003	0.685 ± 0.002	143.2
Classical	14	0.655 ± 0.001	0.617 ± 0.001	0.685 ± 0.001	1.6
No Gates	14	0.654 ± 0.001	0.618 ± 0.002	0.681 ± 0.002	25.9
RY	14	0.655 ± 0.001	0.619 ± 0.003	0.68 ± 0.002	118
RY + RZ	14	0.654 ± 0.001	0.62 ± 0.002	0.684 ± 0.002	179.9
Amp Enc. (Dressed)	15	0.655 ± 0.001	0.618 ± 0.002	0.678 ± 0.002	210.3
Amp Enc. (Undressed)	15	0.665 ± 0.002	0.603 ± 0.002	0.68 ± 0.001	204.9
Classical	15	0.654 ± 0.001	0.62 ± 0.002	0.682 ± 0.001	1.8
No Gates	15	0.654 ± 0.001	0.622 ± 0.001	0.683 ± 0.001	33.8
RY	15	0.654 ± 0.001	0.621 ± 0.004	0.682 ± 0.002	246.2
RY + RZ	15	0.654 ± 0.001	0.618 ± 0.002	0.679 ± 0.002	288.2
Amp Enc. (Dressed)	16	0.653 ± 0.001	0.622 ± 0.002	0.681 ± 0.0	330.6
Amp Enc. (Undressed)	16	0.664 ± 0.002	0.597 ± 0.004	0.679 ± 0.003	299.8
Classical	16	0.655 ± 0.001	0.618 ± 0.002	0.683 ± 0.001	1.6
No Gates	16	0.654 ± 0.0	0.62 ± 0.002	0.682 ± 0.003	49.4
RY	16	0.654 ± 0.001	0.624 ± 0.001	0.681 ± 0.002	291.1
RY + RZ	16	0.656 ± 0.001	0.62 ± 0.003	0.679 ± 0.002	352.8

Classical	17	0.656 ± 0.001	0.616 ± 0.002	0.682 ± 0.001	1.9
No Gates	17	0.654 ± 0.0	0.618 ± 0.002	0.68 ± 0.002	63.5
RY	17	0.654 ± 0.001	0.625 ± 0.002	0.683 ± 0.001	597.7
Classical	18	0.654 ± 0.001	0.618 ± 0.001	0.682 ± 0.001	1.8
No Gates	18	0.655 ± 0.001	0.617 ± 0.002	0.683 ± 0.003	112.8
RY	18	0.655 ± 0.0	0.621 ± 0.002	0.68 ± 0.005	1122.9

B.1 Full Test Graphs

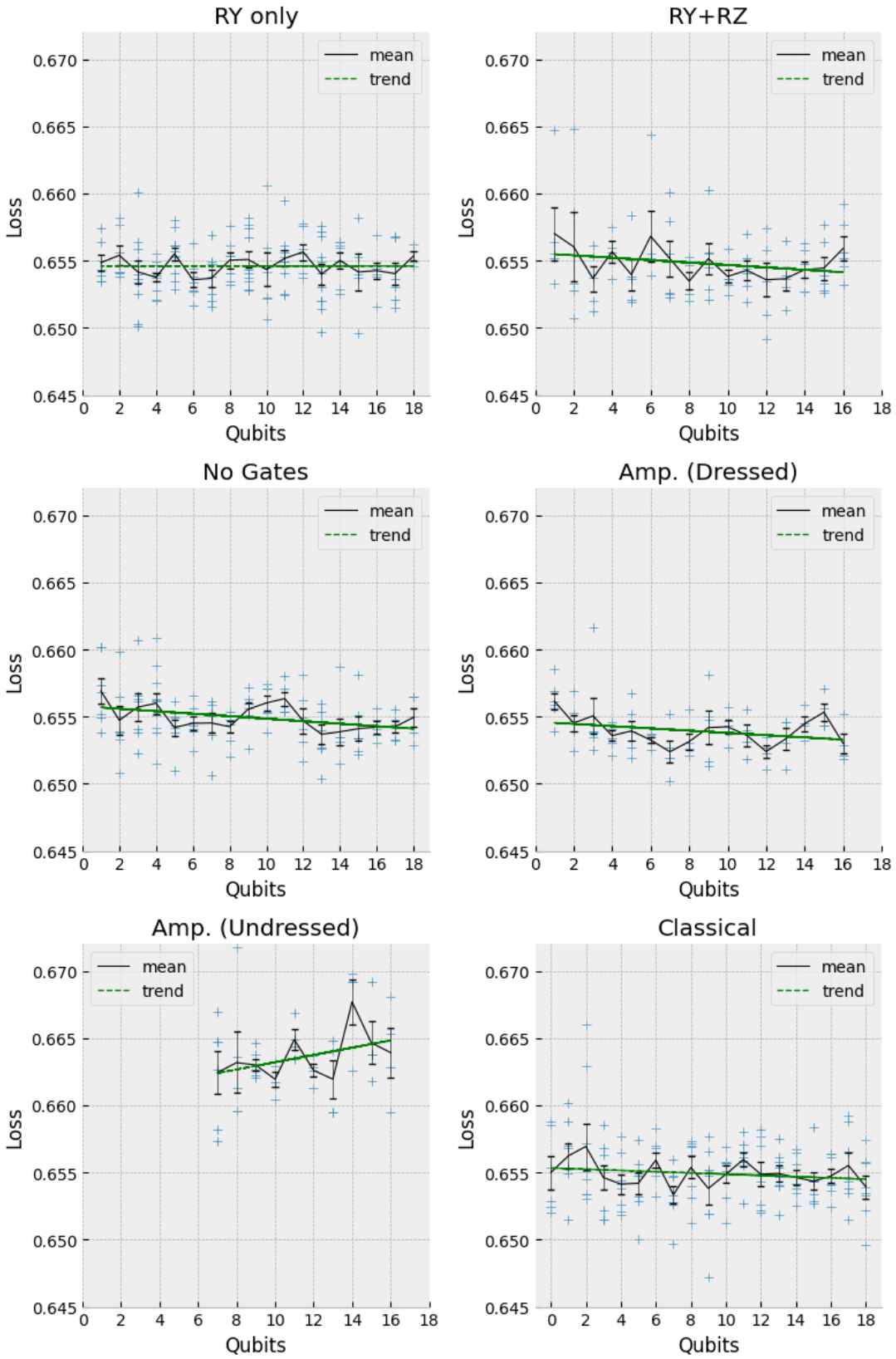

Figure B.1: Test Loss plotted against Qubit Number for all considered circuit designs. Each plot shows datapoints, means and trendline

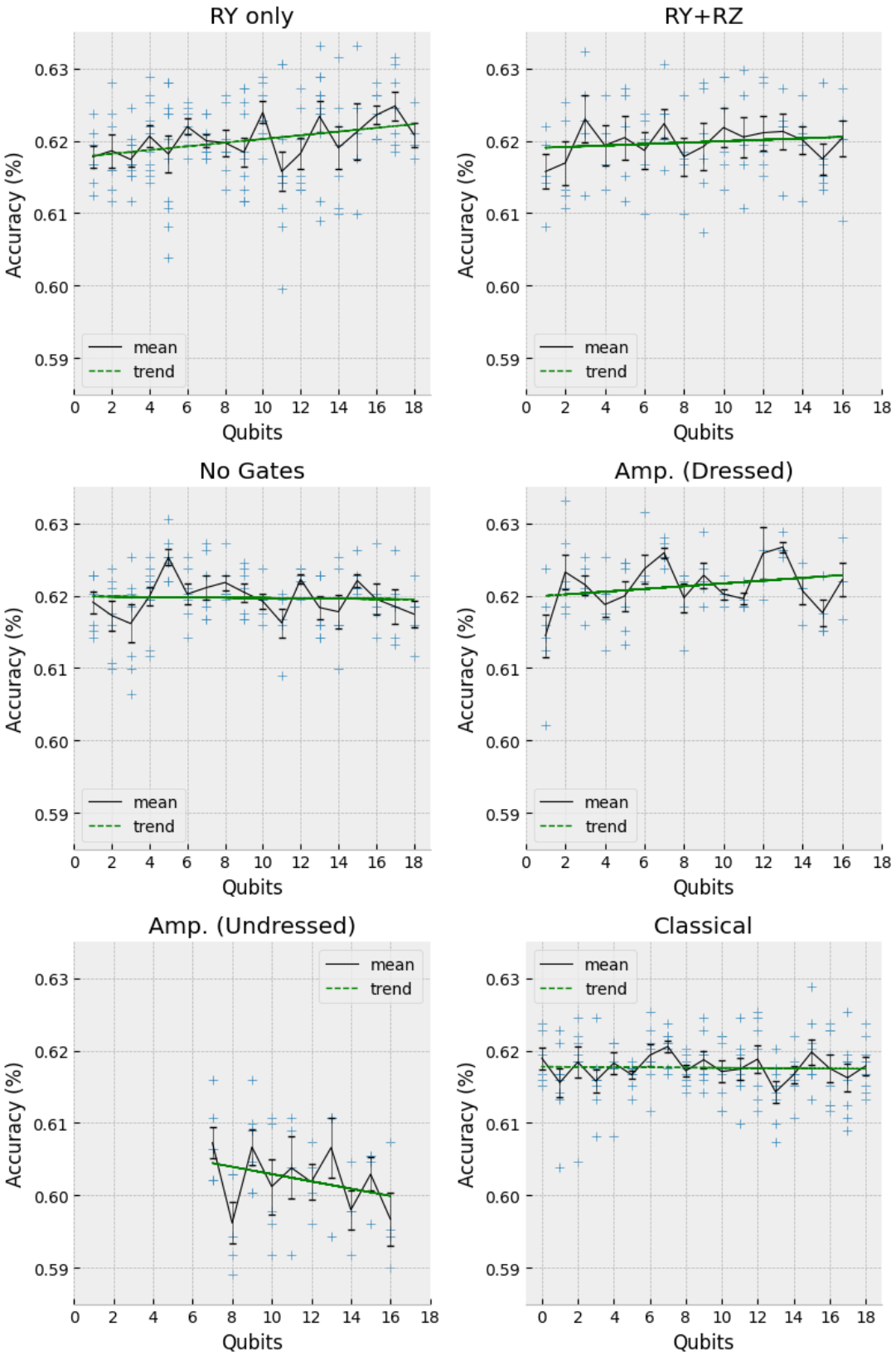

Figure B.2: Test Accuracy plotted against Qubit Number for all considered circuit designs. Each plot shows datapoints, means and trendline

Appendix C

Finetuned BERTbase Performance

Here is included test data for performance of the various circuits when the classical circuit is a finetuned version of BERTbase, provided by Intel (`https://huggingface.co/Intel/bert-`

Type	Qubits	Loss	Acc	f1	
1	Amp Enc. (Dressed)	1	0.5144	0.7924	0.8130
2	Classical	1	0.7534	0.7811	0.8070
3	No Gates	1	0.5401	0.7907	0.8112
4	RY	1	0.6115	0.7863	0.8104
5	RY + RZ	1	0.5526	0.7794	0.8048
6	Amp Enc. (Dressed)	2	0.4978	0.7924	0.8104
7	Classical	2	0.6779	0.7816	0.8073
8	No Gates	2	0.5149	0.7863	0.8101
9	RY	2	0.5947	0.7937	0.8096
10	RY + RZ	2	0.5456	0.7876	0.8108
11	Amp Enc. (Dressed)	3	0.5915	0.7941	0.8019
12	Classical	3	0.6735	0.7760	0.8039
13	No Gates	3	0.5271	0.7889	0.8099
14	RY	3	0.5143	0.7863	0.8092
15	RY + RZ	3	0.4983	0.7855	0.8095
16	Amp Enc. (Dressed)	4	0.4964	0.7846	0.8065
17	Classical	4	0.6401	0.7833	0.8078
18	No Gates	4	0.5341	0.7846	0.8080
19	RY	4	0.5963	0.7910	0.8098
20	RY + RZ	4	0.5088	0.7872	0.8100
21	Amp Enc. (Dressed)	5	0.5491	0.7924	0.8076
22	Classical	5	0.7063	0.7811	0.8073
23	No Gates	5	0.4997	0.7803	0.8061
24	RY	5	0.5301	0.7876	0.8090
25	RY + RZ	5	0.5222	0.7889	0.8099
26	Amp Enc. (Dressed)	6	0.5508	0.7855	0.8059
27	Classical	6	0.6907	0.7833	0.8058
28	No Gates	6	0.5504	0.7889	0.8120
29	RY	6	0.5807	0.7860	0.8075
30	RY + RZ	6	0.5431	0.7898	0.8066
31	Amp Enc. (Dressed)	7	0.5561	0.7872	0.8075
32	Classical	7	0.7286	0.7803	0.8072
33	No Gates	7	0.5802	0.7941	0.8131
34	RY	7	0.5729	0.7924	0.8107
35	RY + RZ	7	0.5864	0.7820	0.8076
36	Amp Enc. (Dressed)	8	0.4949	0.7941	0.8128
37	Classical	8	0.6753	0.7824	0.8046
38	No Gates	8	0.6227	0.7872	0.8098
39	RY	8	0.6028	0.7915	0.8111
40	RY + RZ	8	0.5449	0.7846	0.8086

Appendix D

Mathematical Form of Encoding Only Circuit

Here is set out the brief derivation of the operation equivalent to a quantum circuit with only angle embedding and Pauli Z measurement.

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{D.1}$$

$$\begin{aligned}
&\langle Z\rangle = \langle\psi| Z |\psi\rangle \\
&= \langle 0| HR_Y^\dagger(\theta) ZR_Y(\theta) H |0\rangle \\
&= \langle +| R_Y^\dagger(\theta) ZR_Y(\theta) |+\rangle \\
&= \tfrac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= \tfrac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right) & -2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) \\ -2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) & \sin^2\left(\frac{\theta}{2}\right) - \cos^2\left(\frac{\theta}{2}\right) \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= \tfrac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ -\sin(\theta) & -\cos(\theta) \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= -\sin(\theta)
\end{aligned}$$

Recalling that the input layer converts values to a valid angle by applying $\theta = \left(\frac{\pi}{2}\right)\tanh(x)$, the overall circuit is equivalent to

$$z = -\sin\left(\left(\frac{\pi}{2}\right)\tanh(x)\right)$$

Appendix E

FAdam

When discussing the background section that Hwang 2024 considered the connection between the Adam optimisation and the Fischer Information Matrix, and by doing so suggested an improved version of the optimiser [24]. We include here a comparison of the test performance of the proposed FAdam with the state of the art AdamW provided by Pytorch:

We conclude that, at present, FAdam shows significantly worse performance for optimising Dressed Quantum Circuits.

Type	Loss	Accuracy (%)	F1
RY	0.6546	0.6210	0.6822
RY (FAdam)	0.6618	0.6055	0.6856

Type	Qubits	Loss	Acc	f1	Time
RY	2	0.656	0.619	0.688	5.8
RY (FAdam)	2	0.664	0.603	0.688	9.5
RY	3	0.654	0.617	0.683	8.9
RY (FAdam)	3	0.663	0.602	0.686	18.4
RY	4	0.654	0.621	0.683	10.3
RY (FAdam)	4	0.663	0.604	0.686	15.9
RY	5	0.656	0.620	0.689	13.4
RY (FAdam)	5	0.663	0.605	0.687	26.0
RY	6	0.653	0.624	0.682	16.4
RY (FAdam)	6	0.663	0.607	0.687	16.5
RY	7	0.653	0.620	0.683	25.6
RY	8	0.654	0.622	0.684	24.6
RY (FAdam)	8	0.664	0.602	0.687	24.4
RY	9	0.655	0.618	0.682	28.6
RY (FAdam)	9	0.662	0.604	0.687	33.0
RY	10	0.654	0.622	0.682	32.7
RY (FAdam)	10	0.664	0.601	0.686	32.8
RY	11	0.655	0.614	0.680	37.1
RY (FAdam)	11	0.662	0.606	0.688	37.5
RY	12	0.656	0.618	0.678	44.2
RY (FAdam)	12	0.661	0.607	0.686	43.4
RY	13	0.655	0.622	0.680	51.6
RY (FAdam)	13	0.660	0.606	0.681	48.6
RY	14	0.655	0.621	0.680	75.0
RY (FAdam)	14	0.660	0.612	0.683	76.0
RY	15	0.654	0.629	0.682	166.5
RY (FAdam)	15	0.658	0.611	0.684	113.8
RY	16	0.654	0.626	0.683	211.0
RY (FAdam)	16	0.659	0.606	0.683	181.0