QPMeL: Quantum Polar Metric Learning

by

Vinayak Sharma

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved March 2024 by the
Graduate Supervisory Committee:

Aviral Shrivastava, Chair
Zilin Jiang
Subbarao Kambhampati

ARIZONA STATE UNIVERSITY

May 2024

ABSTRACT

Deep metric learning has recently shown extremely promising results in the classical data domain, creating well-separated feature spaces. This idea was also adapted to quantum computers via *Quantum Metric Learning* (QMeL). QMeL consists of a 2 step process with a classical model to compress the data to fit into the limited number of qubits, then train a *Parameterized Quantum Circuit* (PQC) to create better separation in Hilbert Space. However, on *Noisy Intermediate Scale Quantum* (NISQ) devices, QMeL solutions result in high circuit width and depth, both of which limit scalability. The proposed Quantum Polar Metric Learning (*QPMeL*), uses a classical model to learn the parameters of the polar form of a qubit. A shallow PQC with R_y and R_z gates is then utilized to create the state and a trainable layer of $ZZ(\theta)$-gates to learn entanglement. The circuit also computes fidelity via a SWAP Test for the proposed Fidelity Triplet Loss function, used to train both classical and quantum components. When compared to QMeL approaches, *QPMeL* achieves 3X better multi-class separation, while using only 1/2 the number of gates and depth. *QPMeL* is shown to outperform classical networks with similar configurations, presenting a promising avenue for future research on fully classical models with quantum loss functions.

DEDICATION

To my parents, Sharad & Chithra Sharma, for their unwavering support and encouragement.

To all looking to explore the bounds of human knowledge and have fun doing it.

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my thesis advisor, Dr. Shrivastava, for his active and patient support throughout my Thesis. His encouragement and technical input have been indispensable in this endeavor. Specifically, his help throughout the publication and writing process has helped provide a clear and concise narrative.

Furthermore, I extend my thanks to my labmate, Rishab Kashyap, for his contributions to the experimental work, more specifically executing my code and collecting results, and for serving as a thoughtful sounding board for my ideas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

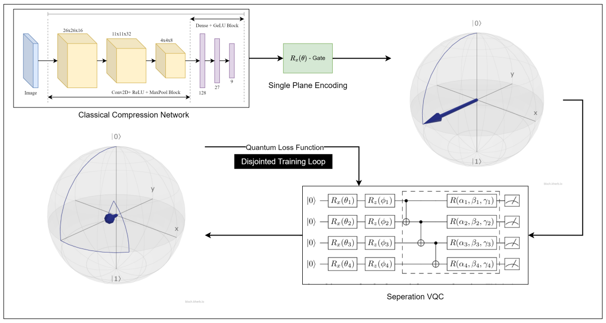
INTRODUCTION

In 2017, Biamonte *et al.* (2017) showed that the ability of Quantum Computers to produce *atypical patterns* which are hard to produce classically, gives them a distinct advantage in the domain of machine learning. However, most devices today are considered *Noisy Intermediate Scale Quantum* (NISQ) devices, which are limited in the circuit breadth (number of qubits) and suffer from high noise at larger circuit depths. Due to this, recent works have focused on creating *Quantum Machine Learning* (QML) models that can be run on NISQ devices.

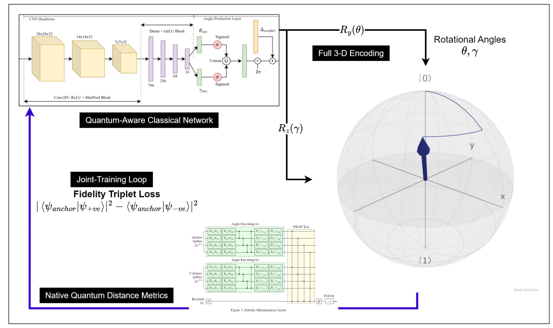
A major challenge in QML is to define an efficient mapping $x \in \mathbb{R}^n \rightarrow |\phi(x)\rangle$ that encodes the classical data into the Hilbert Space(HS). Traditional methods that utilize handcrafted schemes such as *angle encoding* or *amplitude encoding* are space-inefficient (require too many qubits) or require complex circuits. Most critically, all of these schemes demonstrate poor utilization of the Hilbert Space as shown by Lloyd *et al.* (2020).

Quantum Metric Learning (QMeL) (Lloyd *et al.*, 2020) was proposed to address both issues by first compressing the data via a classical method (such as an autoencoder) and then using a Parameterized Quantum Circuit(PQC) to learn a mapping that maps data across the entirety of the associated Hilbert Space. However, this 2-step process is inefficient in the number of operations (gates) performed on the qubits, which for NISQ devices can lead to high noise and errors.

In order to address these challenges, we propose Quantum Polar Metric Learning (*QPMeL*) – a novel method that uses a classical model to learn the parameters of the polar form of a qubit, thereby removing the need for the separation circuit. *QPMeL*



(a) QMeL framework (Lloyd *et al.*, 2020)



(b) Quantum Polar Metric Learning framework (Ours)

Figure 1.1: **QPMeL compared to QMeL**: Figure (a) shows the overview of QMeL and Figure (b) shows the framework of *QPMeL*

creates a more efficient mapping that uses shallower circuits while improving multi-class separability in Hilbert Space. **However, learning the polar form of a qubit classically has 2 main challenges**:

1. Classical distance metrics (such as Euclidean Distance) are not well suited to the curved and complex Hilbert Space. These metrics were formulated for flat real-valued feature spaces.

2. Encoding classical values via a single rotational gate does not sufficiently capture the 3-D nature of a qubit (Bloch Representation) as rotation about a single axis is limited to covering a 2D slice.

Contributions

The purpose of developing *QPMeL* was to address these challenges while best utilizing the strengths of both the classical and quantum computational paradigms. *QPMeL* 's main contributions can be summarized as follows:

1. A novel classical network that encodes classical data into 2 real-valued vectors that are used as Polar coordinates of a qubit. This allows us to utilize the entire 3D space of a qubit, as we are not limited to a single plane.

2. A hybrid Hilbert space distance metric we dub *"Fidelity Triplet Loss"* that measures distance in Hilbert Space while creating the optimization target classically. The distances are measured in-circuit while their difference is computed classically.

3. *Quantum Residual Corrections* to speed up model learning and generate more stable gradients by acting as a noise barrier. The parameters absorb noisier gradients to allow the classical model to learn more efficiently.

The experimental results show that *QPMeL* outperforms the previous QMeL method both in capability and computational complexity. It achieves a high degree of separation between all 10 classes using only 3 qubits and 4 gates per qubit. This is a 3x reduction compared to the 2-layer QAOA implementation of (Lloyd *et al.*, 2020). I also demonstrate that *QPMeL* outperforms classical networks with similar structures to our classical head, presenting a promising avenue for future research on fully classical models with quantum loss functions.

Background

As highlighted in Chapter. 1, one of the limiting challenges in QML is an efficient mapping of classical data to the Hilbert Space. Traditionally, this mapping used handcrafted circuits such as *Basis State Encoding*, *Amplitude Encoding* and *Angle Encoding*. However, basis and angle encoding do not scale with larger data sizes on NISQ devices. While amplitude encoding has exponential scaling (i.e. n qubit can encode 2^n values), it requires normalization constraints as well as complex ansatz searches.

Quantum Feature Space

Unlike classical machine learning where the Feature space is normally characterized by a feature vector $x \in X$, where $x \in \mathbb{R}$, the feature space for *Quantum States* is the *Hilbert Space* ($\mathcal{H}$). This complex vector space is parameterized by the amplitudes of our qubits. The advantage of a Hilbert Space $\mathcal{H}$ is the exponential size, which allows Quantum Computers to efficiently process information and achieve dramatic speedups over classical computers. Additionally, as $\mathcal{H}$ is complex, it is naturally higher dimensional than any $x \in \mathbb{R}$ that it encodes, allowing for natural kernel methods to be applied to it. Both of these ideas are explored in Havlíček *et al.* (2019). However, as demonstrated in Sierra-Sosa *et al.* (2020), the effectiveness of these methods is highly sensitive to the initial state preparation or encoding.

The 3 most common methods for encoding classical data into quantum states are *QRAC*, *Amplitude Encoding*, and *Angle Encoding*. While these are the most common, more specialized encoding schemes have been proposed, and some of those most relevant in *Quantum Machine Learning*(QML) are outlined in Schuld and Petruccione (2018).

Learned Encoding Schemes

A proposed solution to the encoding problem was in the form of *Learned Encoding Schemes* for QML tasks. In this paradigm, a classical computer learns a lower dimensional representation of the data via methods such as Principle Component Analysis(PCA) or Deep Neural Networks(DNNs) and these are then used as the input to a QML circuit. However, as shown by Lloyd *et al.* (2020), the main challenge with this approach is the poor utilization of the Hilbert Space as the classical model creates real-valued nonperiodic bottlenecks that do not translate well to the complex, exponential and periodic natural of Hilbert Space.

Quantum Metric Learning (QMeL)

In order to create embeddings that better utilize the Hilbert space, Lloyd *et al.* (2020) proposed a 2-step procedure of first learning a compressed classical representation and the training a *"Hybrid Bottleneck"* consisting of a PQC (They used the QAOA scheme) and a single classical dense layer to learn better separation. The approach utilized Hilbert Space distance metrics such as *State Fidelity, Helstrøm* or *Hilbert-*

Schmidt to implement a procedure similar to Deep Metric Learning.

Fidelity

is the measure of closeness between two quantum states and can be computed both analytically or via a *"SWAP Test"* circuit. I can define the fidelity between 2 states ρ, ψ as :

$$F(\rho, \psi) = |\langle \rho | \psi \rangle|^2 \tag{1.1}$$

This is analogous to the cosine similarity metric used in classical metric learning as the inner product between states measures a similar notion of similarity.

Chapter 2

RELATED WORK

Metric Learning

Classical metric learning was originally proposed by Chopra *et al.* (2005) via a novel *Contrastive loss* function. Their method was originally designed for classification tasks with a large number of labels that were not available during training. Schroff *et al.* (2015) further extended this idea by incorporating a *Triplet loss* function proposed by Weinberger *et al.* (2005).

Quantum Metric Learning

The idea of Quantum Metric Learning (QMeL) was first proposed Lloyd *et al.* (2020). They suggested first training a classical classifier to serve as a feature extractor. The classifier is then frozen and the classification head is replaced with a linear layer. This layer creates the bottleneck that is then embedded into a Quantum Computer via the QAOA embedding scheme. The linear layer and QAOA layers are then jointly trained to optimize the *State Overlap* that was measured via the Hilbert Schmidt Distance. However, the authors noted that the method suffered from overfitting issues due to the depth of the circuit.

Thumwanit *et al.* (2021) proposed a discrete feature embedding scheme that utilized Quantum Random Access Codes (QRAC). While they also used Quantum Metric Learning, their method was limited to binary features. This is especially clear in their MNIST experiments which downsampled the images to 4×4 and then binarized the pixels. My method does not suffer from similar limitations.

Recently, Liu *et al.* (2022), introduced the idea of *Quantum Few Shot Learning* alongside the *Circuit Bypass Problem* (CBP). Of key interest to this work is the CBP as it allows for a possible method to classically learn efficient quantum mappings. In their paper, the authors define CBP as the tendency of the classical parameters of Hybrid Neural Networks (HyNNs) to learn optimized representations of the dataset without large differences from the Quantum kernel. The proposed cause is that the classical network treats the circuit as a strange non-linearity and optimizes only to get the correct results from the circuit while ignoring its utilization. *QPMeL* aims to exploit this property to produce more stable and efficient embeddings using Triplet Loss in Hilbert Space. However, despite the similarities, I believe that this work cannot be used for direct comparison due to the large differences in the goals of the two papers. Liu *et al.* (2022) focused on solving the few shot task without the same constraints on the circuit depth and breadth as *QPMeL* . This makes setting up a comparison point difficult. Additionally, I believe that the general structure of *QPMeL* can be integrated into Liu *et al.* (2022)'s work.

Nguyen and Chen (2022) proposed an alternate paradigm for embedding search by performing an architecture search on the entanglement operators to find the optimal entanglement pattern. I believe that this work can be used to improve the performance of *QPMeL* by finding the optimal entanglement pattern for the $ZZ(\alpha)$-Gates.

Quantum Triplet Loss

Wendenius *et al.* (2022) also proposed using a parameterized circuit with a classical triplet loss. However, *QPMeL* differs from theirs in several key ways.

1. They use the measurements from a PQC as their embeddings on which they apply Mean Squared Error(MSE)-based triplet loss, I argue that this leads to a loss of information due to the uncertainty principle which guarantees that

measurements cannot be used to fully reconstruct the state. *QPMeL* avoids this issue by computing distances in quantum HS using native distance metrics such as Fidelity.

2. They utilize a classical autoencoder to compress the data, therefore their encoder has no information about the quantum feature space. *QPMeL* trains the system end-to-end, allowing the classical head to learn the quantum feature space.

3. The embeddings are single-dimensional and encoded onto the quantum system using a single layer of R_x gates, this limits access to the entire Bloch sphere surface. *QPMeL* uses a 2D encoding which is used for independent R_y and R_z gates, this allows for access to the entire Bloch sphere surface. Due to these differences, we can also see *QPMeL* greatly outperforms Wendenius *et al.* (2022)'s work by showing strong all-class separation as compared to their work which can only differentiate between 2 classes.

Hou *et al.* (2023) recently proposed a *Quantum Triplet Loss* function. However, there are 2 key differences between their work and *QPMeL* . Firstly, their triplet loss function is fully quantum, and hence the loss is calculated in the circuit. This has implications for the circuit depth and noise resilience. Secondly, their work has a focus on the adversarial properties of the loss function. *QPMeL* does not argue or explore these adversarial properties.

These methods provide a poor point of comparison for *QPMeL* due to either being focused on binary classification or proposing an entirely orthogonal contribution that could be integrated into *QPMeL* .

Quantum Kernel Thoery

Schuld (2021) reformulated the variational learning problem into the problem of finding optimal kernels based on classical *kernel methods*. Schuld in her work proves that all VQC are linear models on the *"feature vectors"* in quantum Hilbert space using the concept of a *Reproducing Kernel Hilbert Space*(RKHS). She then employs the *representer theorem* to show that the optimal kernel can be represented as a linear combination of the training data.

As pointed out by Jerbi *et al.* (2023), her work inspired a line of research where all quantum advantage comparisons against classical methods were only made with respect to the kernel methods. However, they also observed that the *Data Reuploading Classifier* (Pérez-Salinas *et al.*, 2020) breaks the correspondence between the VQC and kernel methods while demonstrating universality on even single qubit systems. As such they proposed a unifying framework for all 3 approaches (VQC, kernel methods, and data reuploading).

Critically, Jerbi *et al.* (2023) highlighted in their results the poor generalization of kernel methods as compared to VQC and data reuploading methods. This was the cornerstone in their argument claiming that the true advantages of quantum machine learning lie in VQC and data reuploading methods. I later show that my work can be interpreted as a method to learn the encoding in kernel space.

Chapter 3

PROPOSED METHOD

The core of *QPMeL* is splitting the learning process between a classical and quantum component as is standard with HyNNs but moving the majority of the learnable parameters to the classical side. This is accomplished by learning the parameters of the polar qubit representation as independent network outputs. *QPMeL* is an extension of the work by Liu *et al.* (2022) and Lloyd *et al.* (2020) and has 4 main components:

1. **Classical Head:** formed by our CNN Backbone and *"Angle Prediction Layer"*,
2. **Quantum Circuits**: Which encodes the classical data and performs the fidelity measurement.
3. **QRC:** An extension to our training circuit that leads to faster training via an additive freely trainable parameter.
4. **Loss Function:** Finally, our loss function is a novel adaptation of the Triplet Loss function to the Hilbert Space. I call this the *"Fidelity Triplet Loss"*.

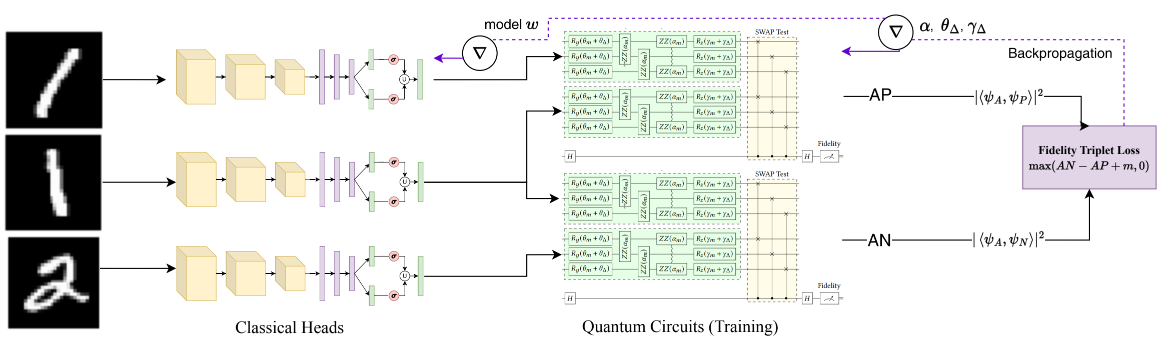


Figure 3.1: ***QPMeL* triplet training loop:** The fidelity triplet loss is computed based on the SWAP test fidelity measurement and the gradients are backpropagated throughout. The classical model weights, ZZ parameters, and QRC parameters are updated together, having the classical head directly learn the polar coordinates that create separation in Hilbert Space.

Classical Head

The classical head has 2 main components:

1. **CNN Backbone:** A standard CNN architecture that learns the features of the input data.
2. **Angle Prediction Layer (APL):** A layer that learns the θ and γ parameters for the quantum circuit.

CNN Backbone

The classical head uses convolution blocks consisting of CONV + ReLU + MaxPool layers, a dense block with 3 Dense + GeLU layers with reducing dimensionality. GeLU has better convergence properties in Dense layers and hence is chosen over ReLU. The architecture can be seen in 3.2.

Angle Prediction Layer (APL)

The polar form of a qubit can be described in terms of 2 angles - θ and γ which can be encoded via the R_y and R_z gates respectively. Due to the rotational nature of these gates, any encoding method using them is periodic. As pointed out by Lloyd *et al.* (2020), when trained together, the classical ReLU learned this periodic property.

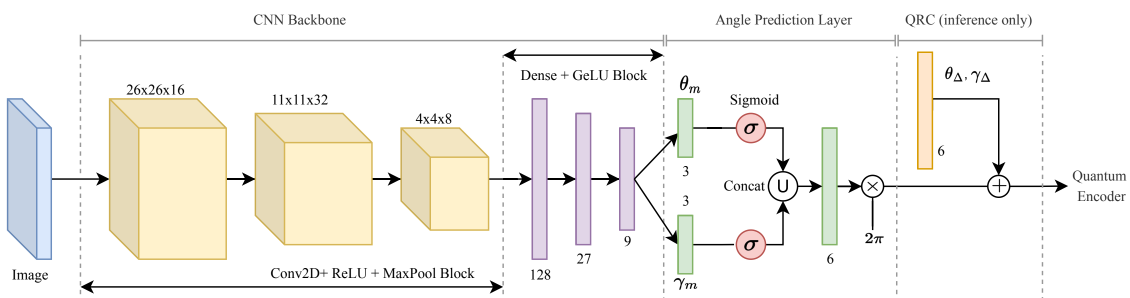


Figure 3.2: ***QPMeL* Classical Head Architecture:** Our angle prediction layer learns independent θ and γ from the features of the CNN Backbone. The sigmoid activation and 2π multiplication are used to match the period of rotation in a qubit. Quantum Residual Correction is then applied for inference.

However, we argue that encoding values directly within the range of the period via a sigmoid and multiply procedure is more efficient.

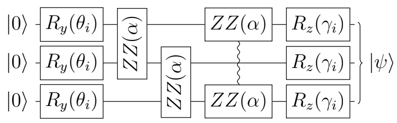
QPMeL aims to learn *"Rotational Representations"* for classical data by creating 2 embeddings for the θ and γ parameters respectively per qubit from the classical head. Therefore for a 3 qubit embedding our classical model would output 6 real values. I ensure that these values are within the period by utilizing the sigmoid activation and multiplying the results by 2π before passing them to the circuit. This explicit period definition ensures more stable losses as we do not need to worry about overlapping values with similar gradients. The idea of rotational embeddings has also been noted in classical networks by Zhou *et al.* (2020), where they demonstrated that rotational representations have continuous representations in 5D and 6D, which lend themselves well to be learned by neural networks. I can hence define the angle prediction layer as follows:

$$
\begin{aligned}
x &= CNN\_Backbone(Image) \\
\theta_m &= sigmoid(W_\theta x + b_\theta) \\
\gamma_m &= sigmoid(W_\gamma x + b_\gamma) \\
APL &= 2\pi \times concat(\theta_m, \gamma_m)
\end{aligned}
$$

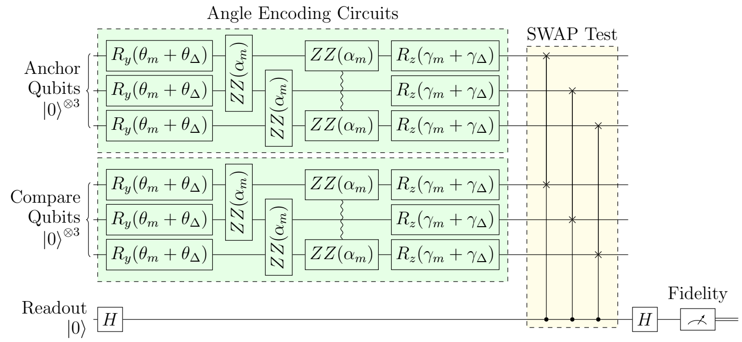
Where, W_θ, b_θ are the weights and bias for the θ prediction layer and W_γ, b_γ are the weights and bias for the γ prediction layer.

Quantum Circuits

There are 2 main settings for our quantum circuits, a training variant and an inference variant. This is mainly to accommodate the swap test required for the fidelity triplet loss 3. I can see both circuits defined in Fig. 3.3.



(a) Encoding (Inference) Circuit



(b) Training Circuit. The green boxes - encoding circuit. The yellow box - SWAP Test implementation. The comparison is either a +ve or -ve sample

Figure 3.3: ***QPMeL* Quantum for training and inference:** The QRC parameters are learned as separate weights during training, acting as noise barriers to the classical head. During inference, the parameters are integrated into the classical head.

Encoding Circuit

The encoding circuit is used to create the state $|\psi\rangle$ from the classical embeddings. The structure (as shown in Fig3.3a) consists of R_y and R_z gates separated by a layer of cyclic $ZZ(\theta)$ gates for entanglement. Our experiments show that this structure performs similarly to or slightly better than an $R_y \rightarrow R_z \rightarrow ZZ$ structure which more naturally shows polar learning. The choice of $ZZ(\theta)$ is motivated by the variable entanglement property as also observed by Lloyd *et al.* (2020).

Embedding State and Learnable Parameters

The final state produced by our Encoding circuit as seen in Fig.3.3a would be:

$$|\psi\rangle = \bigotimes_{i=0}^{n} \exp(i\frac{\phi_i}{2}) \cos\frac{\theta_i}{2} |0\rangle \ + \ \exp(i\frac{-\phi_i}{2}) \sin\frac{\theta_i}{2} |1\rangle \tag{3.1}$$

where,

$$
\begin{aligned}
\phi_i &= \alpha_k - \alpha_i - \gamma_i \\
k &= (n+i) \mod (n+1) \\
\theta_i &= \theta_{m_i} + \theta_{\Delta_i} \\
\gamma_i &= \gamma_{m_i} + \gamma_{\Delta_i} \\
\theta_{m_i}, \gamma_{m_i} &= f(image, w)
\end{aligned}
$$

Where we have 6 parameters per qubit, 2 from the classical model (θ_m, γ_m), 2 learned parameters for the ZZ-Gate (α_i, α_k), and 2 residuals $(\theta_\Delta, \gamma_\Delta)$.

Training Circuit

QPMeL uses separate circuits for training and inference, with 2 main differences - (1). The SWAP test extension requires 2 copies of the encoding circuit (2). Residual Corrections that are only used in the training process. I can see the green regions in Fig.3.3b use our encoding circuit but add the QRC parameters (θ_Δ,γ_Δ) to the rotations.

In order to compute the fidelity we use the SWAP test. I measure $M = P(|0\rangle)$ and convert it to fidelity classically using the formula:

$$F = 2M - 1$$

This structure is seen in the yellow regions alongside the readout qubit on the bottom.

Quantum Residual Corrections (QRC)

I introduce the idea of QRC to speed up the training process and mitigate the smaller gradients that we get due to *Sigmoid Saturation*(Ding *et al.*, 2018) impacting the early layers of our classical model alongside the smaller gradients from our quantum circuit. While using GeLU and ReLU throughout our classical model mitigates the issue of Sigmoid Saturation, it is the combination of the 2 that creates the issue. Quantum gradients are approximated using the *parameter-shift rule* (Mitarai *et al.*, 2018), which utilizes periodic properties of the gates to calculate the gradients, but this also leads to smaller gradients.

To overcome this issue we propose a novel new method we name *"QRC"*. I add learnable parameters θ_Δ and γ_Δ to the angles of the R_y and R_z gates respectively as seen in Fig.3.3. Due to their shallowness and input independence, they are affected by smaller gradients faster and act as *"noise barrier"* allowing our classical to learn faster. During inference, we add these weights to our classical model as seen in Fig.3.2

Fidelity Triplet Loss

QPMeL uses a quantum extension to triplet loss, which uses *State Fidelity* as the distance metric. I simplify our loss function by separating the comparison and distance formulation, favoring 2 calls to a much thinner and shallower circuit as compared to Hou *et al.* (2023). This is more practical on NISQ devices with lower coherence time. *QPMeL* measures distances in Hilbert space using state fidelity and then computes the difference classically. The final loss function in *QPMeL* can be defined as follows:

$$AP = |\langle\psi_A|\psi_P\rangle|^2$$

$$AN = |\langle\psi_A|\psi_N\rangle|^2$$

$$Loss = max(AN - AP + m, 0)$$

Where ψ_A, ψ_P, ψ_N are the quantum states of the Anchor, Positive and Negative samples respectively. m is the margin hyperparameter.

A key difference from the classical counterpart is using $AN - AP$ rather than

$AP - AN$. This is a natural result of the difference between Fidelity and MSE distance metrics. The classical formulation tries to minimize AP Distance, as $MSE(A, P) \to 0$ for similar features. However, in the quantum case, we try to maximize the fidelity, as $F(\psi_A, \psi_P) \to 1$ for similar states.

Chapter 4

ANALYSIS

QPMeL as Dense Angle Encoding

Dense angle encoding is a variation of angle encoding that exploits relative phase as a degree of freedom to encode 2 real features per qubit. Another way for us to frame this would be to consider it a method to encode 2-D data points onto a single qubit as shown by LaRose and Coyle (2020). The structure of *Dense Angle Encoding* is shown below in Eq. 4.1.

$$|p\rangle = \cos(\pi x)\,|0\rangle + e^{2\pi i y}\sin(\pi x)\,|1\rangle \qquad ; p = (x, y) \tag{4.1}$$

A key point to note is that **dense angle encoding essentially encodes data onto to the polar coordinates of a single qubit**. This is apparent when comparing the structure of the polar representation of a single qubit state to the dense angle encoding representation. I can trivially prove that *QPMeL* for single qubits is equivalent to Dense Angle Encoding and hence to Polar Encoding.

Taking our single qubit state as,

$$\begin{aligned} |\psi(\theta,\gamma)\rangle &= \exp(i\frac{-\gamma}{2})\cos\frac{\theta}{2}\,|0\rangle \;+\; \exp(i\frac{\gamma}{2})\sin\frac{\theta}{2}\,|1\rangle \\ &= \exp(i\frac{-\gamma}{2})\left(\cos\frac{\theta_i}{2}\,|0\rangle + \exp(i\gamma)\sin\frac{\theta_i}{2}\,|1\rangle\right) \end{aligned}$$

Without loss of generality, we can consider the global phase of the system as $\exp(i\frac{-\gamma}{2})$, which would reduce our initial mapping to :

$$|\psi(\theta,\gamma)\rangle = \cos\frac{\theta}{2}\,|0\rangle \;+\; e^{i\gamma}\sin\frac{\theta}{2}\,|1\rangle \tag{4.2}$$

Eq. 4.2 is identical to the polar representation of a single qubit state.

Recall that θ and γ are scaled by 2π before encoding into the qubit. Therefore, replacing them in Eq. 4.2 we get:

$$|I\rangle = \cos \pi 2.\sigma(w_\theta I)\,|0\rangle \; + \; e^{i\pi 2.\sigma(w_\gamma I)} \sin \pi 2.\sigma(w_\theta I)\,|1\rangle$$

Where, w_θ, w_γ are the weights from our classical network that yield θ and γ respectively. σ is the sigmoid function and I is the image. Taking $x = 2.\sigma(w_\theta I)$ and $y = \sigma(w_\gamma x)$, we get:

$$|I\rangle = \cos \pi x\,|0\rangle \; + \; e^{2\pi i y} \sin \pi x\,|1\rangle \tag{4.3}$$

Which is identical to the form for dense angle encoding as seen in Eq. 4.1. When considering the multiqubit case, we must also incorporate the entanglement operator $ZZ(\alpha)$ into our state. However, the dense angle encoding structure remains the same with changes in the phase value. A multiqubit formulation of *QPMeL* as a dense encoding would be:

$$|I\rangle = \bigotimes_{i=0}^{n} \cos \pi x_i\,|0\rangle \; + \; e^{-i2\pi\phi_i} \sin \pi x_i\,|1\rangle \tag{4.4}$$

where,

$$\phi_i = \frac{\alpha_k - \alpha_i - y_i}{2\pi} \quad | \quad k = (n+i) \mod (n+1)$$

$$x_i = 2.\sigma(w_\theta^i I) \quad | \quad y_i = 2.\sigma(w_\gamma^i I)$$

This is useful as it allows for *QPMeL* to leverage some of the theoretical properties of dense angle encoding. As shown by LaRose and Coyle (2020), dense angle encoding

1. is more error-resilient than angle encoding,
2. is more resource-efficient than angle encoding,
3. yields higher accuracy as compared to angle encoding, wavefunction/amplitude encoding and superdense angle encoding.

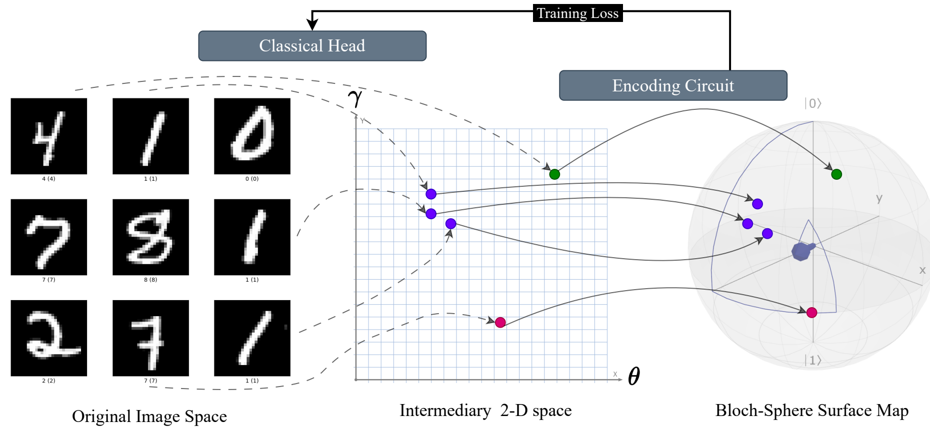


Figure 4.1: **Intermediary Feature Space**: Figure shows the step-by-step mapping between classical and quantum feature spaces. *Note that the classical feature space is not a flat Euclidean feature space but a curved space represented flat for simplicity.*

The key contribution of *QPMeL* over standard angle encoding is coupling our classical encoder to work directly off this paradigm via the split angle embeddings. This creates an *Intermediary Feature Space*, which is curved rather than flat. This is enforced by the sigmoid function and the inherent normality constraints placed on θ and γ when they are encoded onto the Hilbert Space. This is shown in Fig. 4.1.

Bloch Sphere as a Spherical Embedding in QPMeL

The idea of spherical feature spaces in the context of metric learning has been explored by both Wang *et al.* (2017) and Zhang *et al.* (2020). However, while Wang *et al.* (2017) had to implement a novel angular constraint for their triplet loss to create the angular constraints, the use of fidelity distance naturally enforces this constraint on the triplets.

A more interesting line of further study would be to look into applying the work done by Zhang *et al.* (2020) to the *QPMeL* framework. Their team studied the effect of normalization on the gradients and found that proper normalization in the classical space can lead to better convergence. This is particularly interesting as the normalization of the qubit states is a natural part of the *QPMeL* framework. Their proposed modification to the triplet loss function normalized the embeddings before computing the distances:

$$L_{Triplet} = (||\hat{f}_a - \hat{f}_p||^2 - ||\hat{f}_a - \hat{f}_n||^2 + m)_+$$
$$where, \quad \hat{f} = \frac{f}{||f||^2}$$

The analog for *QPMeL* is:

$$L_{\text{Fidelity Triplet}} = (|\langle\psi_a|\psi_p\rangle|^2 - |\langle\psi_a|\psi_n\rangle|^2 + m)_+$$

In the *QPMeL* framework, our distance metric (fidelity) operates on quantum states which are normalized by nature. This would imply that it would inherit the desirable properties of the spherical embeddings. However, to prove that these properties hold for the gradients generated by parameterized quantum circuits would require further study and be beyond the scope of this work.

QPMeL as a Kernel Learner

As Schuld (2021) proved, the space of quantum models is mathematically identical to the *reproducing kernel Hilbert space* (RKHS) of Quantum Kernels. In this context, we can consider our classical head a *kernel learner* similar to the one proposed by Le and Xie (2018).

The proof and associated kernel structures can be seen below:

QPMeL *Kernel Function*

Using the definition of Quantum Kernel defined by Schuld (2021),

$$k(x, x') = |\langle\phi(x)|\phi(x')\rangle|^2 \tag{4.5}$$

where,

$\phi =$ mapping function from classical $\rightarrow$ quantum,

$k =$ kernel function

This is by design identical to the state fidelity due to the function of a kernel as a *similarity metric*. As "x" in the embedding from classical space, for *QPMeL* , we can define the embedding function as,

$$|\phi(x)\rangle = |\psi(\theta, \gamma)\rangle \tag{4.6}$$

Therefore, the kernel function in the *QPMeL* framework is parameterized by θ, γ defined as,

$$k(x, x') = |\langle\psi(\theta, \gamma) \mid \psi(\theta', \gamma')\rangle|^2 \tag{4.7}$$

The *QPMeL* data encoding feature map $\psi(\theta, \gamma)$ was defined in Eqn. 3.1. This embedding function is significantly more complex than the embedding defined by a single rotational embedding considered by Schuld (2021). For the state of simplicity, let us only consider the rotational embedding components while ignoring the entanglement $ZZ(\alpha)$ operator (as would be the case when $\alpha = 0$).

$$\psi(\theta, \gamma) = \bigotimes_{i=0}^{n} e^{-i\frac{\gamma_i}{2}} \cos\frac{\theta_i}{2}|0\rangle \; + \; e^{i\frac{\gamma_i}{2}} \sin\frac{\theta_i}{2}|1\rangle$$

In order to better understand the kernel function, let us consider the case of a single qubit. As Schuld (2021) formulated the kernel function in terms of density matrices, I will be using the density matrice version of QPMeL.

$$|\psi(\theta, \gamma)\rangle = e^{-i\frac{\gamma}{2}} \cos\frac{\theta}{2}|0\rangle \; + \; e^{i\frac{\gamma}{2}} \sin\frac{\theta}{2}|1\rangle \tag{4.8}$$

The density matrix for our single qubit encoded state would be:

$$\rho(x) = |\psi(\theta,\gamma)\rangle \langle\psi(\theta,\gamma)| =$$
$$\cos^2\frac{\theta}{2}|0\rangle\langle 0| \ + \ e^{i\gamma}\cos\frac{\theta}{2}\sin\frac{\theta}{2}|0\rangle\langle 1| + e^{-i\gamma} \ \sin\frac{\theta}{2}\cos\frac{\theta}{2}|1\rangle\langle 0| + \ \sin^2\frac{\theta}{2}|1\rangle\langle 1|$$

Therefore the inner product of two such states would yield our kernel function,

$$k((\theta,\gamma),(\theta',\gamma')) = t^2 + \omega^2 + 2t\omega\cos(\gamma - \gamma') \tag{4.9}$$

where,

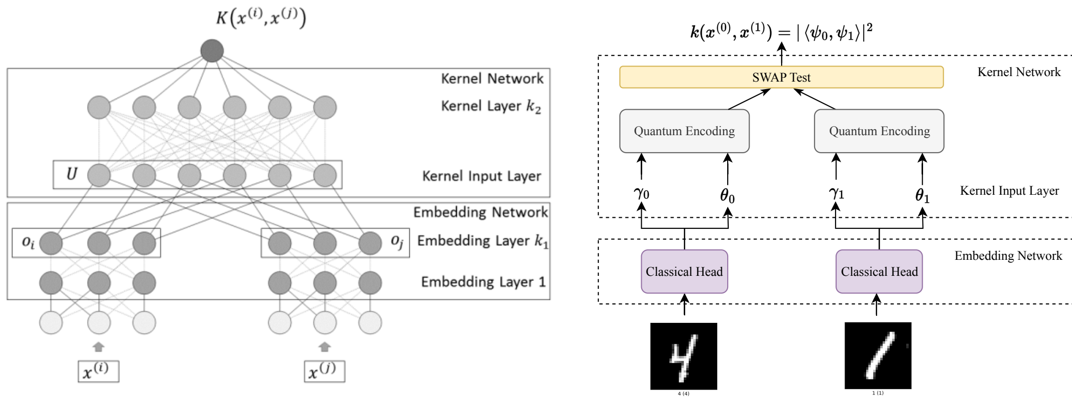
$$t = \cos\frac{\theta}{2}\cos\frac{\theta'}{2} \tag{4.10}$$
$$\omega = \sin\frac{\theta}{2}\sin\frac{\theta'}{2} \tag{4.11}$$

This is quite similar to the kernel function of the linear model that Schuld (2021) uses. However, their paper used 1-D feature vectors alongside multiple parameters for the rotational gates forming the VQC. In contrast, *QPMeL* has 2-D feature vectors and only a single set of parameters for variational ZZ gates.

QPMeL *as a Deep Embedding Kernel*

As mentioned at the beginning of Section. 4, we can consider the *QPMeL* as an extension of the Deep Embedding Kernel network introduced by Le and Xie (2018). This can be justified by looking at the structural similarities between the two frameworks as seen in Fig. 4.2. We can further formalize the similarities between the two frameworks as follows:

1. The output of the Kernel network from Le and Xie (2018) is the probability that the inputs belong to the same class. In *QPMeL* the output of our training circuit is the fidelity between 2 encoded states which is also the probability that the inputs belong to the same class.



(a) Deep Kernel Embedding (Le and Xie (2018), Fig. 1)

(b) *QPMeL* in similar structure

Figure 4.2: **QPMeL as DEK**: From Fig.(a) and (b) we can clearly note the similar structure of the two frameworks with analogous components.

2. The embedding network in DEK(Le and Xie, 2018) creates optimal lower dimensional representations of the data based on the loss from the kernel network. In *QPMeL* , the classical head performs the same function.

Based on these parallels, we can define *QPMeL* as a Deep Embedding Quantum Kernel(DEQK) with the simplified kernel defined in Eq. 4.9. This would allow us to leverage the theoretical foundations of kernel methods while having the data efficiency and expressiveness of a deep network.

Chapter 5

EXPERIMENTS

In order to quantify the degree of separation across multiple feature spaces without relying of classification as the only use case, we require a new metric.

MinMax Metric:

I propose the *"MinMax"* metric, a feature space agnostic metric to quantify our embedding performance.

The metric computes the distance of the AP_{max} and AN_{min}, which corresponds to the worst case for both and represents it as a % AN_{min}. These are computed on the averages seen in the heatmap to avoid outliers. However, as the metric trends are reversed for MSE and Fidelity, we present the results separately. Additionally, we use $1 - distances$ for Fidelity so that the sign implications remain consistent for MSE and Fidelity.

$\mathbf{AP}_{max}$ Worst case for Anchor-Positive distance.

- *(MSE)* The maximum distance between the same class.
- *(Fidelity)* The minimum distance between the same class.

$\mathbf{AN}_{min}$ Worst case for Anchor-Negative distance.

- *(MSE)* The minimum distance between different classes.
- *(Fidelity)* The maximum distance between different classes.

Positive values indicate that we can draw a decision boundary between the classes while negative values indicate that the classes are not separable with this feature space. Additionally, the magnitude of the value indicates the degree of separation.

Classical Baselines:

I trained 3 models with the same structure as our classical model in Fig.3.2 but different activations on the final layer - **(1). Sigmoid Model** with sigmoid activation,**(2). Scaled Sigmoid model** adds 2π scaling to (1) and **(3). ReLU model** with ReLU activation. All were trained with MSE Triplet Loss. I compare them against the Classical Head before applying QRC (till 2π multiplication in Fig.3.2).

Quantum Baselines:

I establish 3 Quantum Baselines - **(1). No Residual Model** which is identical to *QPMeL* but without QRC, **(2). QMeL**(Lloyd *et al.*, 2020) a lightly modified version of the original using our Fidelity triplet loss and **(3). QMeL+**, a heavily modified version of the original using MSE Triplet Loss for pre-training alongside our Fidelity Triplet Loss.

Comparable Methods:

As mentioned in Chapter. 2, most of the comparable methods are limited to binary classification. Our proposed method

Setup

All experiments were carried out using the *'lightning.qubit'* and *'default.tf'* backends on pennylane for our quantum simulations. The MNIST dataset was used as a benchmark. I use the *"All Pair Distances"* for visualization which are plotted as heatmaps. I randomly take 1000 samples for each ordered pair, the results of which are averaged to create the heatmaps. I use a margin(m) of 0.9 for all experiments.

Chapter 6

RESULTS

I divide our results into 2 sections: *Performance Analysis* and *Efficiency Analysis*. The former focuses on the performance of our model in terms of its ability to learn a decision boundary and the latter focuses on the computational complexity of our model.

Performance Analysis

Table 6.1: **MinMax Metric, Euclidean Space:** Sigmoid models cannot create decision boundaries. The remaining models create a decision boundary. *QPMeL* models perform the best

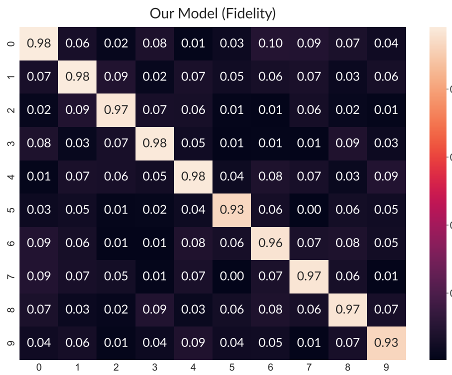
Model	AP_{max}	AN_{min}	Diff	% of AN_{min}
QPMeL Model	0.141	1.143	1.002	87.685
QPMeL Classical Head	0.003	0.029	0.026	88.303
ReLU Model	1.161	4.605	3.445	74.800
Sigmoid Model	0.193	0.012	-0.181	-1474.017
Scaled Sigmoid Model	3.644	1.935	-1.709	-88.331

(Higher is Better)

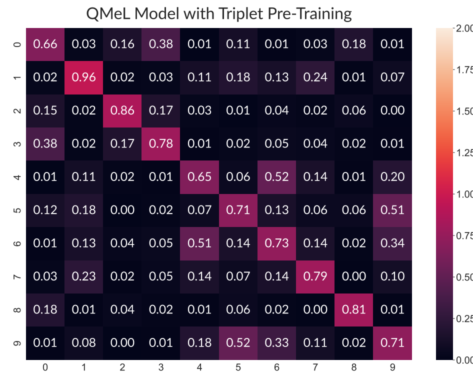
Quantum-Classical Coupling

Despite never being trained on any Euclidean loss function such as MSE-Triplet Loss, the *QPMeL* classical head produces a strong separation between classes as seen by the 88% of AN_{min} in Table 6.1. I can also see in Fig.6.2a that this separation is well distributed across all pairs.

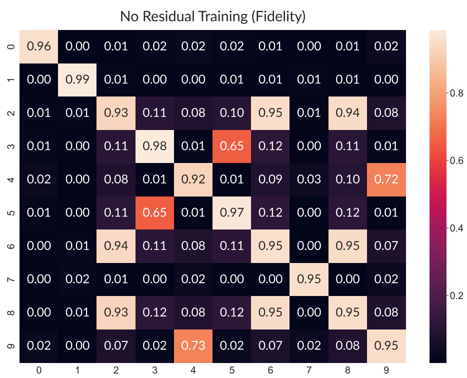
This is because, as is clear from Eqn 3.1, the state produced by the encoding circuit is directly dependent on the classical model outputs of θ, γ. Therefore to learn



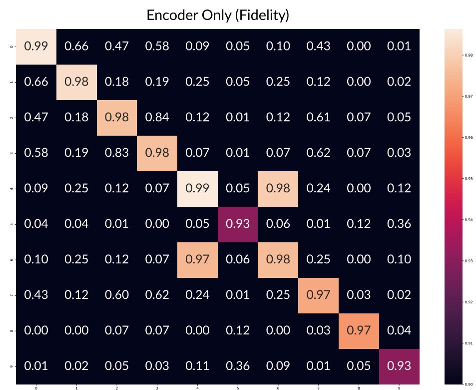
(a) *QPMeL* Model
(Perfect Separation)



(b) QMeL+ Model
(Smaller Perfect Separation)



(c) No Residual Model
(Multiple Outliers)



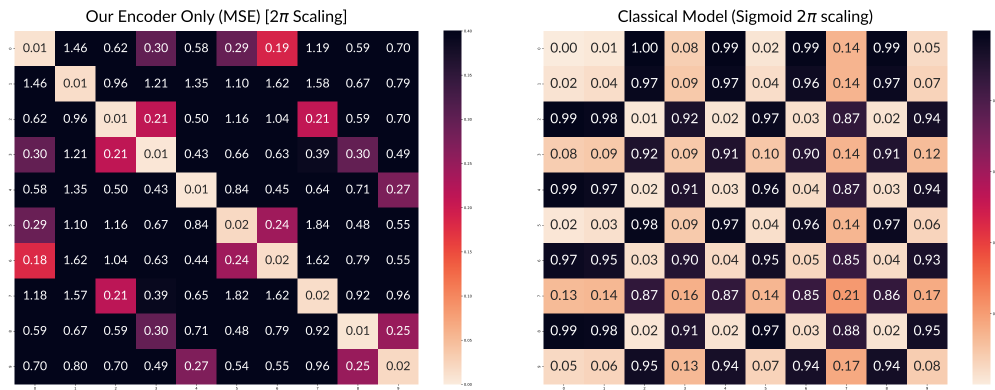
(d) *QPMeL* Classical Head Only
(4-6 pair is not separable)

Figure 6.1: **All Pair heatmaps in Hilbert Space:** (a) shows our model fully trained with residuals, with perfect separation and large differences. (b).shows a modified version of the QMeL model with perfect separation but smaller differences between positive and negative classes. (c). shows the same model trained without residuals with multiple outliers. (d). only creates binary separation for the digit 9

separability in Hilbert Space, our loss function enforces separation in Euclidean Space due to the no overlap guarantee of the 2π output scaling.

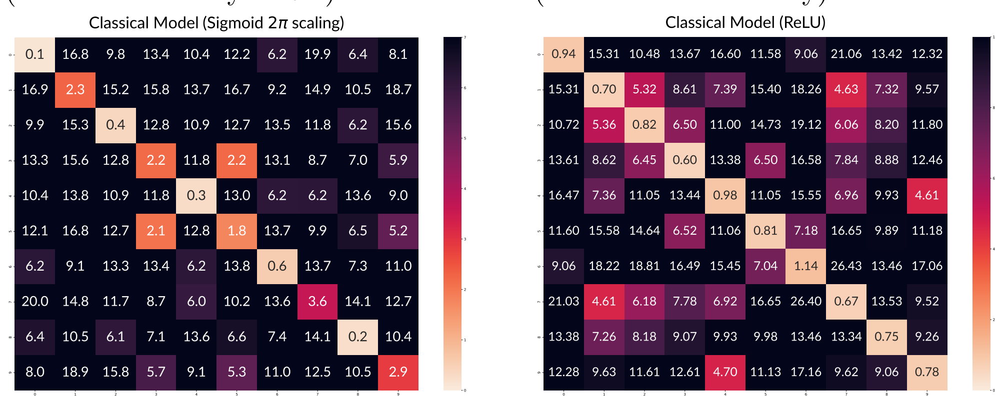
Classical Models with Quantum Loss

Look at Table 6.1, both sigmoid models produce negative differences which indicate the absence of a decision boundary. The *QPMeL* classical head has an identical structure to the **Scaled Sigmoid model** with only a difference in the loss function.



(a) *QPMeL* Classical Head (Decision Boundry at 0.1)

(b) Sigmoid Activation (No Decision Boundry)



(c) Scaled Sigmoid model (No Decision Boundry)

(d) ReLU Activation (Decision Boundry at 1.5)

Figure 6.2: **All Pair heatmaps in Euclidean space:** (a) and (c) have the same model architecture but show large differences in separatability. (b) shows perfect separatability but via an unbounded upper limit as seen by the magnitude of values.

However, it produces a positive difference of 88% of AN_{min}, implying that a strong decision boundary exists.

As both models are capable of learning the same family of functions due to identical model architectures, the difference is indicative that our quantum loss function allows the classical head to learn a better metric function. Additionally, the *QPMeL* classical head also shows an 18% improvement over the ReLU model (Table 6.1), proving that the improvement is substantial. As noted by Liu *et al.* (2022) in their CBP, the classical network treats our circuit as an unknown non-linearity which we

Table 6.2: **MinMax Metric for Hilbert Space:**both the original and the no residual models cannot create a decision boundary, our model creates a strong separation when compared to the other approaches.

Model	AP_{max}	AN_{min}	Diff	% of AN_{min}
QPMeL Model	0.075	0.906	0.831	91.681
No Residual Model	0.092	0.051	-0.041	-81.363
QMeL Model	0.489	0.018	-0.470	-2579.563
QMeL+ Model	0.363	0.483	0.120	24.882

(Higher is Better)

argue is benefiting the learning capability of the classical head. I believe that this is a promising avenue for future research, which can be explored further by using the *QPMeL* framework for other classical tasks.

Quantum Residual Correction Impact

From the results in Table. 6.2, we can see that the No Residual model fails to learn a clear decision boundary. However, looking Fig.6.1c, while the model fails to create a decision boundary, the issues are localized to specific pairs (ex. (8,2)), while all other regions remain well separated. I believe this is due to smaller gradients which require more training time and data to learn these corner cases. However, PQCs are known to have barren plateaus which can make training unstable.

In contrast, our model can learn a decision boundary for all classes. Fig.6.1d, shows that even before applying our corrections the encoder struggles with a single pair. This implies that QRC is helping the classical model learn the corner cases faster. This is further supported by the fact that the QRC is only used during training and not during inference. This hints that the residual framework eases the task on the classical head allowing it to learn faster and more robustly.

Figure 6.3: **Outliers by Max Samples:** Heatmaps show fewer outliers as 'k' values increase. (a) k=1 (0.001%), (b) k=5 (0.005%), (c) k=15 (0.015%), (d) k=25 (0.025%)

Outlier Analysis

Fig. 6.3, shows the number of outliers as we increase the worst-case sample size. I define the *worst-case samples* as the top-k values for pairwise fidelity. As highlighted in Section 5, we sample 1000 random data points per digit-digit pair. This yields a $(10 \times 10 \times 1000)$ tensor. We then calculated top-k values for each pair and averaged them to get the final value.

We can clearly see that even for the average of the top 15 samples which constitutes the top 0.015% of sampled data points, the number of outliers is negligible. This is a strong indicator that our model is robust to outliers.

Efficiency Analysis

When compared to the original QMeL Paper, we use 1/2 the number of gates and 1 layer circuit depth to achieve 3x better separation. The original QMeL framework (Lloyd *et al.*, 2020) as explained in Section. 2 used an overlap loss that did not scale well to multi-class tasks. The modified version with our Fidelity Triplet Loss also failed to create a decision boundary as seen in Table.6.2 where it produces a negative difference. When we apply a more robust separation method for the pre-quantum step **(QMeL+)** the difference in Table.6.2 is positive implying that a decision boundary can be made but the magnitude is only 24% of AN_{min} implying that the clusters are close together.

In contrast, *QPMeL* produces a positive difference that is 91% of AN_{min}, showing 3x improvement over QMeL+. I see this trend reflected in Fig.6.1a and Fig.6.1b with the difference in magnitude between the diagonal and everywhere else. Additionally, looking at the parameters in Table 6.3 *QPMeL* utilizing 1/2 (9 vs 21) the number of gates, 1/2 (5 vs 11) the circuit depth and (11k vs 16k)20% fewer classical parameters.

Table 6.3: **Computation complexity comparison:** Our model uses (1/2) the number of gates and (1/2) the circuit depth and 50% fewer classical parameters compared QMeL with better performance.

Model	# of Gates	Circuit Depth	Classical Parameters
QMeL/QMeL+	21	11	16,645
QPMeL Model	9	5	11,099

Chapter 7

CONCLUSION

PQCs and QML as an extension present a promising new avenue for research. However, the limitation of current hardware makes near-term applications difficult to realize. In this paper, we propose the *QPMeL* framework that learns the polar representation of qubits via Hilbert Space Metric Learning. I also introduce the idea of QRC which helps alleviate the issues of sigmoid saturation and barren plateaus. Our results present a promising new direction for research utilizing PQCs as loss functions or non-linear activations to enhance classical models and show strong representation learning.

REFERENCES

Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd, "Quantum machine learning", Nature **549**, 7671, 195–202 (2017).

Chopra, S., R. Hadsell and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification", in "2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)", vol. 1, pp. 539–546 vol. 1 (2005).

Ding, B., H. Qian and J. Zhou, "Activation functions and their characteristics in deep neural networks", in "2018 Chinese Control And Decision Conference (CCDC)", pp. 1836–1841 (2018).

Havlíček, V., A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces", Nature **567**, 7747, 209–212, URL `https://doi.org/10.1038%2Fs41586-019-0980-2` (2019).

Hou, Y.-Y., J. Li, X.-B. Chen and C.-Q. Ye, "Quantum adversarial metric learning model based on triplet loss function", (2023).

Jerbi, S., L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel and V. Dunjko, "Quantum machine learning beyond kernel methods", Nature Communications **14**, 1, URL `https://doi.org/10.1038%2Fs41467-023-36159-y` (2023).

LaRose, R. and B. Coyle, "Robust data encodings for quantum classifiers", Physical Review A **102**, 3, URL `http://dx.doi.org/10.1103/PhysRevA.102.032420` (2020).

Le, L. and Y. Xie, "Deep embedding kernel", (2018).

Liu, M., J. Liu, R. Liu, H. Makhanov, D. Lykov, A. Apte and Y. Alexeev, "Embedding learning in hybrid quantum-classical neural networks", in "2022 IEEE International Conference on Quantum Computing and Engineering (QCE)", (IEEE, 2022), URL `https://doi.org/10.1109%2Fqce53715.2022.00026`.

Lloyd, S., M. Schuld, A. Ijaz, J. Izaac and N. Killoran, "Quantum embeddings for machine learning", (2020).

Mitarai, K., M. Negoro, M. Kitagawa and K. Fujii, "Quantum circuit learning", Physical Review A **98**, 3, URL `http://dx.doi.org/10.1103/PhysRevA.98.032309` (2018).

Nguyen, N. and K.-C. Chen, "Quantum embedding search for quantum machine learning", IEEE Access **10**, 41444–41456, URL `https://doi.org/10.1109%2Faccess.2022.3167398` (2022).

Pérez-Salinas, A., A. Cervera-Lierta, E. Gil-Fuster and J. I. Latorre, "Data re-uploading for a universal quantum classifier", Quantum **4**, 226, URL http://dx.doi.org/10.22331/q-2020-02-06-226 (2020).

Schroff, F., D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in "2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", (IEEE, 2015), URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

Schuld, M., "Supervised quantum machine learning models are kernel methods", (2021).

Schuld, M. and F. Petruccione, *Supervised learning with quantum computers*, vol. 17 (Springer, 2018).

Sierra-Sosa, D., M. Telahun and A. Elmaghraby, "Tensorflow quantum: Impacts of quantum state preparation on quantum machine learning performance", IEEE Access **8**, 215246–215255 (2020).

Thumwanit, N., C. Lortaraprasert, H. Yano and R. Raymond, "Trainable discrete feature embeddings for variational quantum classifier", (2021).

Wang, J., F. Zhou, S. Wen, X. Liu and Y. Lin, "Deep metric learning with angular loss", (2017).

Weinberger, K. Q., J. Blitzer and L. Saul, "Distance metric learning for large margin nearest neighbor classification", in "Advances in Neural Information Processing Systems", edited by Y. Weiss, B. Schölkopf and J. Platt, vol. 18 (MIT Press, 2005), URL https://proceedings.neurips.cc/paper_files/paper/2005/file/a7f592cef8b130a6967a90617db5681b-Paper.pdf.

Wendenius, C., E. Kuehn and A. Streit, "Training parameterized quantum circuits with triplet loss", in "Joint European Conference on Machine Learning and Knowledge Discovery in Databases", pp. 515–530 (Springer, 2022).

Zhang, D., Y. Li and Z. Zhang, "Deep metric learning with spherical embedding", (2020).

Zhou, Y., C. Barnes, J. Lu, J. Yang and H. Li, "On the continuity of rotation representations in neural networks", (2020).