

TYPHOON TRAJECTORY FORECASTING VIA QUANTUM-ENHANCED PARAMETER EFFICIENT LEARNING

Chen-Yu Liu¹ Kuan-Cheng Chen²

¹ National Taiwan University, Taipei, Taiwan

² Imperial College London, London, UK



Technical Project Report

"Forecasting the Future with Quantum-Enhanced Intelligence."

— Anonymous

ABSTRACT

TYPHOONS are among the most devastating natural disasters, causing significant loss of life, infrastructure damage, and economic disruption worldwide. Accurate typhoon trajectory forecasting is crucial for disaster preparedness and mitigation, yet remains computationally challenging due to the complexity of atmospheric dynamics and the need for large-scale deep learning models. Traditional forecasting models require extensive computational resources, making them less accessible and energy-intensive. Quantum-centric supercomputing provides a promising framework for integrating the strengths of quantum and classical computing to tackle such large-scale challenges. However, Quantum Machine Learning (QML), despite its theoretical advantages, faces practical limitations—including efficient data encoding and reliance on quantum hardware for inference—which hinder its scalability for real-world applications such as climate forecasting. To address these challenges, we introduce Quantum-Train (QT), a novel QML paradigm that leverages quantum neural networks (QNNs) to generate trainable parameters for classical models exclusively during training, eliminating the need for quantum hardware at inference time. Building on QT’s success across multiple domains—including image classification, reinforcement learning, LSTM-based flood forecasting (which won second place in the Deloitte Quantum Climate Challenge), and fine-tuning billion-parameter large language models (LLMs)—we propose applying QT to the critical task of typhoon trajectory forecasting. As part of the Pascal Quantum Challenge, we develop Quantum Parameter Adaptation (QPA), an extension of QT designed for parameter-efficient learning in large-scale models. By integrating QPA with a deep learning-based Attention-based Multi-ConvGRU model, we demonstrate a computationally efficient and scalable solution for typhoon prediction. **Moreover, this work is the first to demonstrate the feasibility of QML for the highly complex task of typhoon trajectory forecasting, marking a significant advancement in quantum-enhanced climate modeling.** Additionally, we explore the compatibility of QT and QPA with neutral atom quantum computing, particularly through **Pasqal’s Qadence Python package**, paving the way for future quantum acceleration of climate applications. Our results show that QPA significantly reduces the number of trainable parameters while maintaining model accuracy, leading to lower energy consumption, and increased accessibility of high-performance forecasting models. This work advances the real-world applicability of hybrid quantum-classical learning techniques, demonstrating their potential to revolutionize climate forecasting with greater efficiency, scalability, and sustainability.

CONTENTS

1	Introduction	5
1.1	Problem Statement	5
1.2	Sustainability Relevance	5
1.3	AI & Quantum Integration	5
2	Mathematical Formulation & Solution Strategy	7
2.1	Mathematical Model	7
2.1.1	Quantum-Train	7
2.1.2	Parameter-Efficient Fine-Tuning Methods	9
2.1.3	Quantum Parameter Adaptation	9
2.2	Solution Approach	10
3	AI Contribution & Methodology	11
3.1	Techniques: Classical AM-ConvGRU	11
3.2	Data & Processing	11
4	Quantum Computing Integration & Feasibility	12
4.1	Theoretical Perspective of Quantum-Train-based Methods	12
4.1.1	Quantum-Train Recap	13
4.1.2	Error Performance Analysis	13
4.2	Compatibility	15
4.2.1	Digital Quantum Computing	15
4.2.2	Digital-Analog Quantum Computational Paradigm	15
4.3	Energy Efficiency	16
4.3.1	Via Parameter Reduction in Quantum-Train	16
4.3.2	In the Context of Pasqal's Neutral Atom Quantum Computing	17
5	Implementation, Code & Testing	17
5.1	Project Architecture	17
5.2	MWE & Testing	17
6	Results, Evaluation & Roadmap	18
6.1	Performance & Benchmarks	18
6.1.1	Overall Benchmarking	18
6.1.2	On Hyperparameters of QPA	19
6.1.3	Resulting Trajectories Visualization	20
6.2	Deployment Challenges	21
6.2.1	Compatibility	22
6.2.2	Challenge of Measurement	22

6.3	Carbon Footprint Analysis	22
6.3.1	Runtime Estimation	22
6.3.2	Estimate Emission from Runtime	23
6.4	Future Outlook	24
6.4.1	Leveraging the Potential of Neutral Atom Quantum Computing	24
6.4.2	Beyond Climate Forecasting	25
A	QT Approximation Error	29

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Climate change has intensified the frequency and severity of typhoons, leading to devastating consequences for communities worldwide. The increased frequency of extreme weather events underscores the urgent need for more accurate trajectory forecasts to mitigate human and economic losses (Emanuel, 2005; Webster et al., 2005; Masson-Delmotte et al., 2021). Accurate forecasting allows for timely disaster response, evacuation planning, and resource allocation, significantly reducing casualties and infrastructure damage. Taiwan, where Team QTX comes from, characterized by its steep and mountainous terrain, is particularly vulnerable to extreme weather events. On average, the island experiences approximately 3.5 typhoons and dozens of torrential rainstorms each year, resulting in annual economic losses of **374.3 million** Euros due to infrastructure damage, agricultural losses, and disruptions to economic activity (Kuo et al., 2022). The high frequency and intensity of these events highlight the critical need for advanced forecasting models to improve disaster preparedness and resilience.

Traditional computational approaches to weather prediction have made substantial progress (Bauer et al., 2015), but they often struggle to model the inherent complexity of typhoon dynamics. These challenges arise due to the chaotic nature of atmospheric systems, the high-dimensional parameter spaces, and the need for extensive computational resources to process vast amounts of meteorological data. Current numerical weather prediction (NWP) models rely on large-scale supercomputing infrastructure, which is costly and energy-intensive. As both data volume and model sophistication continue to increase, these limitations become more pronounced, necessitating more efficient and scalable forecasting methodologies.

A key challenge in modern weather forecasting is the high computational cost associated with training complex models. Large-scale meteorological models require extensive parameter tuning and optimization, leading to:

- **High energy consumption:** Training deep learning models for weather forecasting demands significant computing power, contributing to high operational costs and carbon emissions.
- **Limited accessibility:** Developing countries and resource-constrained research institutions may lack access to the supercomputing infrastructure needed for training state-of-the-art forecasting models.
- **Scalability concerns:** As weather models grow in complexity, their training becomes increasingly resource-intensive, limiting their deployment for real-time forecasting applications.

Thus, parameter-efficient training methods are essential for enhancing the sustainability, accessibility, and scalability of advanced weather forecasting models. Reducing the number of trainable parameters while maintaining model accuracy provides several advantages: (1) it accelerates model convergence, enabling faster updates and adaptation to evolving weather patterns; (2) it lowers hardware requirements, allowing sophisticated forecasting models to be trained and implemented across a wider range of computational resources; and (3) it improves generalization, as efficient models are less susceptible to overfitting, thereby enhancing resilience to shifting climate conditions.

1.2 SUSTAINABILITY RELEVANCE

With the increasing intensity of typhoons due to climate change, there is a growing need for sustainable forecasting solutions. Traditional numerical weather prediction models are computationally intensive, requiring large datasets and significant energy consumption from high-performance computing systems. By improving the process of typhoon trajectory forecasting via Quantum-Enhanced method, Quantum-Train (QT) (details in the following content), this project directly contributes to climate resilience and early warning systems, crucial for mitigating the impacts of extreme weather events. QT enhances parameter efficiency, reducing the computational cost and time required for accurate training and predictions. Moreover, leveraging a quantum-classical hybrid computing approach leads to energy-efficient training while maintaining high accuracy. This work aligns with sustainability goals by reducing resource consumption and improving disaster preparedness.

1.3 AI & QUANTUM INTEGRATION

Quantum computing emerges as a promising paradigm that can address high-dimensional optimization and state-space exploration with unprecedented efficiency (Montanaro, 2016). By integrating quantum-enhanced

methods with classical computing—an approach termed quantum-centric supercomputing (Bravyi et al., 2022; Gambetta, 2022)—these complementary strengths alleviate computational bottlenecks. Decomposing complex problems into components suited for classical or quantum computing allows for more efficient problem-solving schemes. Classical systems are well-suited for tasks like data processing, while quantum computing shows potential in optimization and exploring large state spaces.

Conventional quantum machine learning (QML) approaches employ parameterized quantum circuits (PQCs) as quantum neural networks (QNNs) (Chen et al., 2020), where data is input through specific data encoding methods (Pérez-Salinas et al., 2020; Schuld et al., 2021). The updates to QNN parameters during the training process are computed on the classical side, creating a hybrid quantum-classical computing framework (Mari et al., 2020). While QML offers improvements in specific applications (Cerezo et al., 2022; Huang et al., 2022; Biamonte et al., 2017; Caro et al., 2022; Huang et al., 2021), significant challenges remain—particularly in data encoding for large datasets.

One such proposal to address both the data encoding challenge and the requirement for quantum hardware during the inference stage is Quantum-Train (QT) (Liu et al., 2024b). Instead of using a QNN to interact directly with the data, this approach leverages QNNs to generate the weights of a target classical neural network model during the training process. The QT method: (1) Keeps the data input process entirely within the classical model. (2) Eliminates the need for quantum computing hardware during inference, as the trained model is purely classical. (3) Reduces the number of training parameters on a polylogarithmic scale by mapping quantum state basis to the target neural network parameters ($M \rightarrow \text{polylog}(M)$). Current studies on QT have demonstrated promising results across various domains, including image classification (Liu et al., 2024b), reinforcement learning (Liu et al., 2024c), and long short-term memory (LSTM) models for flood prediction (Lin et al., 2024). Notably, the LSTM-based flood prediction work won second place in the Deloitte’s Quantum Climate Challenge 2024¹. QT effectively reduces the number of trainable parameters while maintaining task performance. Importantly, the trained model remains fully classical, enabling deployment on classical computers without requiring quantum hardware for inference.

As a further extension from QT, Quantum Parameter Adaptation (QPA) is proposed in top AI venue ICLR 2025 (Liu et al., 2025b). QPA utilize the concept of QT –leverage the Hilbert space to do the parameter compression—but further extend the concept from directly compressing the machine learning (ML) model to compressing the parameter-efficient fine-tuning parameters (Hu et al., 2021; Liu et al., 2024d; Housby et al., 2019; Lin et al., 2020; Li & Liang, 2021; Yang et al., 2021). That is, consider a low-rank adaptation (LoRA) for a LLM, two low-rank matrices $A \in \mathbb{R}^{l \times r}$ and $B \in \mathbb{R}^{r \times k}$ are generated to represent the update $AB = \Delta W \in \mathbb{R}^{l \times k}$ of a given weight matrix $W \in \mathbb{R}^{l \times k}$, with the rank $r \ll \min(l, k)$. QPA uses QT to generate the parameters of A and B , thus it is possible to use extremely small amount of parameters (in the scale of $\text{polylog}(lr + rk)$) to fine-tune large language models (LLMs).

Drawing inspiration from these successes—particularly in flood prediction and LLM fine-tuning—we propose extending QT to tackle climate challenges that require large ML models, specifically typhoon trajectory forecasting. This task is known for its high computational demands, yet we believe that the latest advancements in QT offer a more efficient and scalable solution.

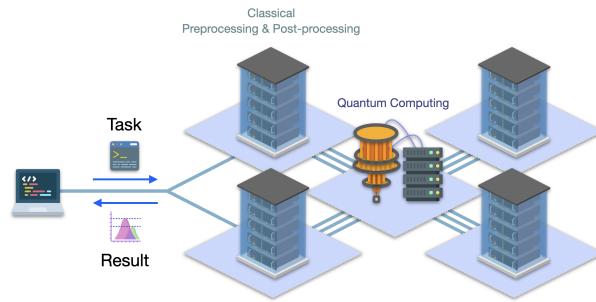


Figure 1: Quantum-centric supercomputing scheme.

¹Deloitte’s Quantum Climate Challenge 2024: <https://www.deloitte.com/de/de/issues/sustainability-climate/deloitte-quantum-climate-challenge.html>

2 MATHEMATICAL FORMULATION & SOLUTION STRATEGY

2.1 MATHEMATICAL MODEL

In this section, we first present the mathematical foundation of the QT method, followed by an extended version known as Quantum Parameter Adaptation.

2.1.1 QUANTUM-TRAIN

To begin with the QT approach, we first define the training parameters of a neural network (NN) for a given task—regardless of its specific architecture at this stage—as $\boldsymbol{a} = (a_1, a_2, \dots, a_m)$, where m represents the total number of parameters in the NN.

We consider the QT process that provides a different approach from conventional QML. A PQC with $N = \lceil \log_2 m \rceil$ qubits and L layers is constructed using a circuit ansatz, represented as:

$$|\psi(\boldsymbol{\theta})\rangle = \left(\prod_{i=1}^{N-1} \text{CNOT}^{i,i+1} \prod_{j=1}^N R_Y^j(\theta_j^{(L)}) \right)^L |0\rangle^{\otimes N}, \quad (1)$$

where the single-qubit rotation gate R_Y^j is associated with the tunable parameter $\theta_j^{(L)}$, with qubit index j and layer index L , and CNOT represents the two-qubit controlled-NOT gate. With a Hilbert space of size 2^N , where $2^N \geq m$, this PQC produces 2^N distinct measurement probabilities, $|\langle \phi_i | \psi(\boldsymbol{\theta}) \rangle|^2 \in [0, 1]$ for $i \in \{1, 2, \dots, 2^N\}$. The parameter size of $\boldsymbol{\theta}$ depends on N and L , where L is a hyperparameter similar to those in classical ML. Typically, L scales proportionally to the number of qubits, either $O(N)$ or in some cases $O(N^2)$ (Cerezo et al., 2021; Sim et al., 2019; Benedetti et al., 2019), though it can be generalized to a looser polynomial scale, $O(\text{poly}(N))$. Thus, with polynomial layers in the PQC, we can generate $2^{\lceil \log_2 m \rceil} \geq m$ parameters (probabilities) using $O(\text{polylog}(m))$ PQC parameters.

At this stage, the measurement probabilities are values between 0 and 1. To map these probabilities to the target parameters $\boldsymbol{a} \in \mathbb{R}^m$, we employ a MLP mapping model G with tunable parameters \boldsymbol{b} . The input to G is the binary representation of the basis (of length N) and the corresponding measurement probability, such that

$$G_{\boldsymbol{b}}(|\phi_i\rangle, |\langle \phi_i | \psi(\boldsymbol{\theta}) \rangle|^2) = a_i, \quad \forall i \in \{1, 2, \dots, m\}. \quad (2)$$

Here, only the first m basis states are used to cover all parameters in the target NN. Since the input size of $G_{\boldsymbol{b}}$ is $N + 1$, the size of \boldsymbol{b} can also be controlled at a scale of $O(\text{polylog}(M))$. Consequently, \boldsymbol{a} is generated from the output of the PQC and the mapping model $G_{\boldsymbol{b}}$. By tuning $\boldsymbol{\theta}$ and \boldsymbol{b} , we effectively influence the value of the loss function \mathcal{L} , which is evaluated by the target NN for a given task. A comparison of conventional QML and QT is shown in Fig. 2.

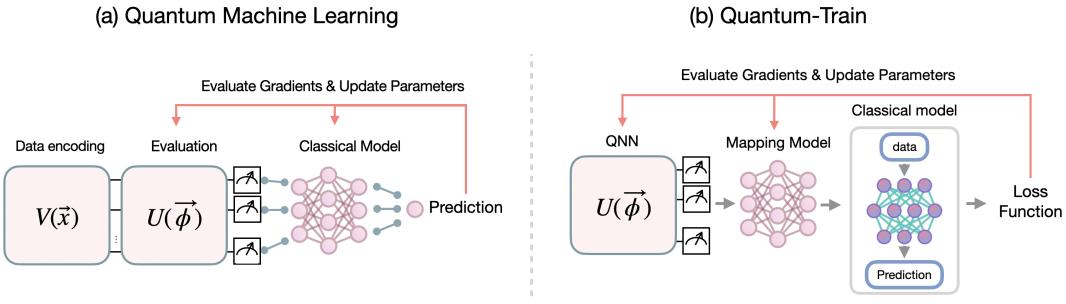


Figure 2: Overview of (a) Quantum Machine Learning (Mari et al., 2020; Mitarai et al., 2018), (b) Quantum-Train (Liu et al., 2024b)

Gradient Estimation of Quantum Circuit Compressed Parameters. The target NN parameters \boldsymbol{a} are generated through the use of a PQC coupled with a mapping model. The quantum-dependent parameters, represented as $(\boldsymbol{\theta}, \boldsymbol{b})$, impact the target NN parameters via the quantum state preparation and measurement steps.

The gradient of the loss function, reflecting the influence of quantum parameters, is expressed as:

$$\nabla_{\theta, b} \mathcal{L} = \left(\frac{\partial \mathbf{a}}{\partial (\theta, b)} \right)^T \cdot \nabla_{\mathbf{a}} \mathcal{L}. \quad (3)$$

In this expression, $\frac{\partial \mathbf{a}}{\partial (\theta, b)}$ denotes the Jacobian matrix, which describes how sensitive the classical parameters \mathbf{a} are to changes in the quantum parameters (θ, b) . This provides a high-level overview of the gradient in an exact quantum state simulation. In practical applications involving real quantum computers, the gradient calculation must account for the parameter shift rule and its variants (Mitarai et al., 2018; Schuld et al., 2019).

Parameter Update of Quantum Circuit Compressed Parameters. The learning rate η is a critical factor, particularly due to the complex dynamics introduced by the quantum-classical interface. The update for the quantum parameters is defined as:

$$\theta_{t+1}, b_{t+1} = \theta_t, b_t + \eta \nabla_{\theta, b} \mathcal{L}. \quad (4)$$

This rule ensures that the quantum parameters are updated to optimize the performance of the target NN.

Using the gradient computation and parameter update rules discussed earlier, the QT parameter generation process has been successfully applied to various domains, including image classification with convolutional neural networks (CNNs) (Liu et al., 2024b;a; Liu & Chen, 2024), flood prediction (time series) with LSTM models (Lin et al., 2024), and policy gradient reinforcement learning in CartPole and MiniGrid environments (Liu et al., 2024c). These applications have demonstrated a significant reduction in trainable parameters while maintaining comparable performance, establishing QT as a general framework for using QML as a parameter-efficient learning approach.

Batched Parameter Generation Strategy. If the target NN model has m parameters, the required number of qubits can be determined as $N = \lceil \log_2 m \rceil$. For example, scaling up to $m = 10^9$ (one billion) parameters would necessitate $N = 30$ qubits. While a quantum system with 30 qubits is feasible with current hardware and classical simulations, the associated GPU memory demand (approximately 16 GB) and computational overhead (several seconds per circuit evaluation) (Google, 2024) pose significant challenges for machine learning tasks that require frequent iterations within a practical timeframe.

To address this, our approach (Liu & Chen, 2024) partitions the m parameters of the target NN into n_{ch} chunks, where each chunk comprises n_{mlp} parameters, ensuring that $n_{ch} = \lceil m/n_{mlp} \rceil$. This effectively reduces the required qubit count by mapping multiple parameters of the target NN to a single quantum basis.

The mapping model, denoted as \tilde{G}_b , receives $|\phi_i\rangle$ and $|\langle\phi_i|\psi(\theta)\rangle|^2$ as inputs and generates a batch of n_{mlp} parameters \mathbf{a} for each chunk as output.

$$\mathbf{a} = (\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_{n_{ch}}), \quad (5)$$

$$\tilde{G}_b(|\phi_i\rangle, |\langle\phi_i|\psi(\theta)\rangle|^2) = \tilde{a}_i, \quad \forall i \in \{1, 2, \dots, n_{ch}\}, \quad (6)$$

$$\tilde{a}_i = (a_{i,1}, a_{i,2}, \dots, a_{i,j}), \quad \forall j \in \{1, 2, \dots, n_{mlp}\}. \quad (7)$$

This is accomplished by employing a decoder-like structure within the MLP of the mapping model \tilde{G}_b , where the output size is expanded from 1 to n_{mlp} . The detailed configuration used in this work is presented in Table 1. Consequently, the required number of qubits is reduced from $N = \lceil \log_2 m \rceil$ to

$$N = \lceil \log_2 n_{ch} \rceil = \lceil \log_2 \left(\lceil \frac{m}{n_{mlp}} \rceil \right) \rceil. \quad (8)$$

This modification effectively lowers the qubit requirement by approximately $\lceil \log_2 n_{mlp} \rceil$ qubits compared to the original QT approach. The original method can be considered a special case where $n_{mlp} = 1$. While this reduction in qubit count increases the number of parameters in the mapping model due to the expanded output size, it provides a substantial memory advantage, as the storage required for the quantum state decreases by a factor of $1/n_{mlp}$.

For instance, when $m = 10^9$ and $n_{mlp} = 1024$, the required qubit count can be reduced to

$$N = \lceil \log_2 \left(\lceil \frac{10^9}{1024} \rceil \right) \rceil = 20. \quad (9)$$

This represents a 33% reduction in qubit usage, lowering the memory needed to store the quantum state to just 1/1024 of that required for 30 qubits, while also improving the efficiency of classical simulations.

Table 1: Configuration of the mapping model \tilde{G}_b , with N representing the number of qubits for each task.

Hyperparameter	Meaning	Value
Input size	Input of the mapping model ($ \phi_i\rangle, \langle\phi_i \psi(\theta)\rangle ^2$)	$N + 1$
Hidden dimension	Main structure of the MLP mapping model	$[32, 32, n_{mlp}]$

2.1.2 PARAMETER-EFFICIENT FINE-TUNING METHODS

Before introducing QPA, it is essential to first discuss Parameter-Efficient Fine-Tuning (PEFT) methods to provide a smoother transition to the topic.

PEFT methods are designed to mitigate the computational challenges of fine-tuning LLMs by significantly reducing the number of trainable parameters while maintaining, or even enhancing, model performance. Among the most prominent approaches are LoRA (Hu et al., 2021) and DoRA (Liu et al., 2024d), which leverage the assumption that weight updates during fine-tuning reside within a low-rank subspace. Rather than modifying the full weight matrices, these methods introduce small, trainable low-rank decomposition matrices, effectively reducing the number of trainable parameters while still capturing crucial model adaptations.

Another class of PEFT techniques employs adapters—lightweight feed-forward layers inserted between existing layers of the neural network (Houlsby et al., 2019; Lin et al., 2020). During fine-tuning, only these adapter layers are trained, while the rest of the model remains unchanged, significantly lowering computational costs. Meanwhile, Prefix Tuning (PT) (Li & Liang, 2021; Yang et al., 2021) follows a different paradigm by introducing learnable prefix vectors that are prepended to input or hidden states at each layer. During fine-tuning, only these prefix vectors are updated, leaving the core model parameters frozen. By concentrating fine-tuning efforts on a small subset of parameters, these methods optimize computational efficiency while preserving task-specific adaptability, making them well-suited for large-scale language models.

2.1.3 QUANTUM PARAMETER ADAPTATION

Beyond the batched parameter generation strategy, the fundamental distinction between QT and QPA lies in the modification of the training target. In QPA, instead of learning the full set of model parameters for a target neural network, the focus shifts to learning only the parameters of a PEFT method. This adjustment enables QPA to handle significantly larger-scale tasks with greater efficiency. Moreover, by incorporating batch parameter generation, the scalability of the target task is further extended. In this framework, the parameter vector a now corresponds to the parameters of a PEFT method.

For instance, considering LoRA as an example, a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ undergoes a low-rank decomposition for its update, expressed as: $W_0 + \Delta W = W_0 + BA$ where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$. In this case, QPA generates these two low-rank matrices using Eq. 6 and Eq. 7, where a represents the elements of A and B . The required number of qubits in the LoRA scenario is then given by:

$$N = \lceil \log_2 \left(\lceil \frac{r(d+k)}{n_{mlp}} \rceil \right) \rceil. \quad (10)$$

Consider a case where $W_0 \in \mathbb{R}^{2048 \times 1024}$, resulting in a weight matrix with 2 million parameters. By setting $r = 4$, the low-rank decomposition gives $B \in \mathbb{R}^{2048 \times 4}$ and $A \in \mathbb{R}^{4 \times 1024}$. Given a chunk size of $n_{mlp} = 64$, the required number of qubits is calculated as:

$$N = \lceil \log_2 \left(\lceil \frac{4(2048+1024)}{64} \rceil \right) \rceil = 8. \quad (11)$$

In contrast, applying the standard QT method directly would require 21 qubits, demonstrating the efficiency of QPA in significantly reducing qubit usage. The fine-tuning process in QPA follows the gradient evaluation and update rules outlined in Eq. 3 and Eq. 4. This framework can be extended to other parameter-tuning tasks, as demonstrated in (Liu et al., 2025b). A schematic representation of QPA is provided in Fig. 3.

In cases where gradient-based tuning is not applicable, alternative non-gradient-based optimization methods, such as Nelder-Mead, SPSA, or COBYLA, may be employed for updating parameters effectively.

Quantum Parameter Adaptation (ICLR 2025)

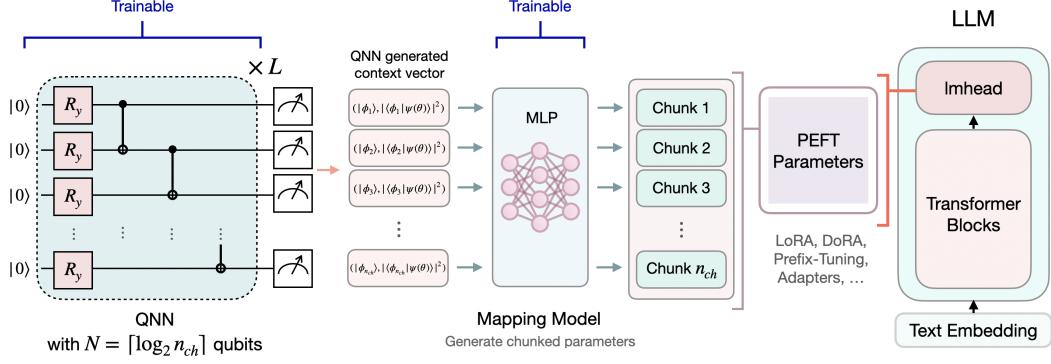


Figure 3: Overview of Quantum Parameter Adaptation (Liu et al., 2025b).

2.2 SOLUTION APPROACH

With an understanding of QPA, we can now outline our strategy for leveraging QPA to train a typhoon trajectory forecasting model using real historical typhoon trajectory data. This leads to the central topic: **Typhoon Trajectory Forecasting via Quantum-Enhanced Parameter-Efficient Learning**.

To begin, we investigate classical ML approaches for typhoon trajectory forecasting and identify a prior study that achieves strong performance using an Attention-based Multi-ConvGRU model (Xu et al., 2022). With 8 million parameters, this model strikes a balance between complexity and feasibility, making it an ideal candidate for enhancement through QPA within a reasonable timeframe—specifically, the 3 to 4 weeks of the Pascal Quantum Challenge. By applying QPA to this model, we aim to validate its effectiveness as a test case, showcasing the computational benefits and practical impact of quantum-enhanced parameter-efficient learning. Building on this foundation, we defer the details of the classical approach to the next section, while a graphical overview of our solution strategy is presented in Fig. 4.

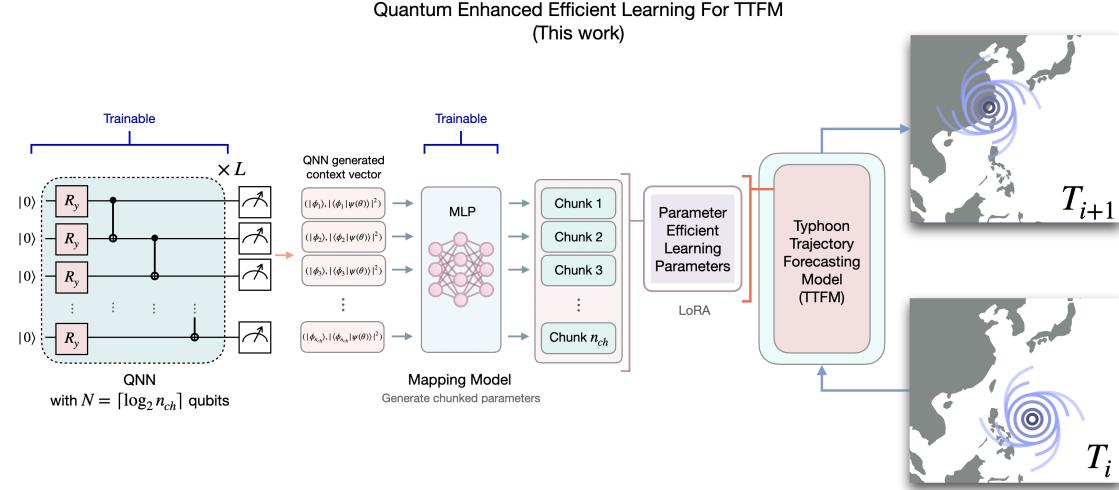


Figure 4: Overview of our solution strategy in this work.

3 AI CONTRIBUTION & METHODOLOGY

3.1 TECHNIQUES: CLASSICAL AM-CONVGRU

In this section, we outline the key elements of the classical ML method for typhoon trajectory prediction, which serves as the foundation for applying QPA to this task. While a detailed explanation of the classical model is beyond the scope of this work, readers can refer to the original paper ([Xu et al., 2022](#)) for a comprehensive discussion. The primary objective of this project is to demonstrate that, regardless of the complexity of the underlying classical model—constructed with numerous intricate mechanisms, as discussed in the previous and following sections, QPA enables training with an exceptionally small number of parameters. This claim will be validated in the results section.

The Attention-based Multi-ConvGRU (AM-ConvGRU) model is a deep learning-based approach designed to predict typhoon trajectories by effectively integrating spatial and temporal features. The key motivation behind AM-ConvGRU is the necessity to capture both 2D meteorological features from traditional typhoon records and 3D atmospheric features that influence the evolution of typhoons across multiple isobaric levels. To achieve this, the model employs a Wide & Deep framework that consists of a Generalized Linear Model (GLM) and a deep learning-based AM-ConvGRU network.

The deep learning component of the model is built on Convolutional Gated Recurrent Units (ConvGRU), which serve as an alternative to ConvLSTM, reducing the computational cost while effectively capturing spatio-temporal dependencies. Additionally, a Residual Channel Attention Block (RCAB) is introduced to adaptively select high-response isobaric planes, enhancing the model’s ability to identify critical pressure levels that significantly influence typhoon movements. The RCAB mechanism improves feature selection across different atmospheric layers, allowing the model to focus on the most relevant meteorological patterns.

The Wide Component, represented as a GLM, extracts traditional 2D typhoon-related features such as latitude, longitude, wind speed, and central pressure. These features provide a statistical basis for trajectory prediction. Meanwhile, the Deep Component, consisting of AM-ConvGRU, extracts nonlinear spatial and temporal dependencies from 3D reanalysis data, modeling the atmospheric evolution of typhoons. The final trajectory forecast is obtained by fusing both components.

The ConvGRU cell, which is the core of the deep component, is defined as follows:

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t, \quad (12)$$

where h_t represents the hidden state at time t , z_t is the update gate, and \tilde{h}_t denotes the candidate activation state, computed as:

$$\tilde{h}_t = f(W_x * x_t + W_h * (r_t \circ h_{t-1}) + b). \quad (13)$$

Here, $*$ denotes the convolution operation and \circ represents the Hadamard product, r_t is the reset gate, and b is the bias term. The update gate z_t and reset gate r_t are computed as:

$$z_t = \sigma(W_z * x_t + U_z * h_{t-1} + b_z), \quad (14)$$

$$r_t = \sigma(W_r * x_t + U_r * h_{t-1} + b_r), \quad (15)$$

where σ is the sigmoid activation function. By leveraging ConvGRU, the model effectively captures spatial correlations while maintaining sequential memory, enabling it to model typhoon trajectories with greater accuracy. Here, we present the fundamental equations and core concepts of the AM-ConvGRU model, while the full details can be found in the original paper ([Xu et al., 2022](#)).

3.2 DATA & PROCESSING

The AM-ConvGRU model is trained by QPA and evaluated using real-world typhoon datasets sourced from the China Meteorological Administration (CMA) ([Ying et al., 2014](#)) and the ERA-Interim reanalysis dataset provided by ECMWF ([Molteni et al., 1996](#)). The CMA dataset contains historical typhoon records, including 2D meteorological features such as latitude, longitude, wind speed, and central pressure. Meanwhile, the ERA-Interim dataset provides 3D atmospheric reanalysis data at multiple isobaric levels (e.g., 1000 hPa, 750 hPa, 500 hPa, 250 hPa), offering a more detailed representation of typhoon structures. The visualization of typhoon paths of the CMA dataset is shown in Fig. 5.

To prepare the data for model training, the CMA dataset undergoes preprocessing using the CLIPER ([Xu et al., 2022](#)) method, which extracts structured feature vectors from raw typhoon records. For the ERA-Interim dataset, the Earth’s surface is partitioned into a $1^\circ \times 1^\circ$ grid, and a 31×31 spatial window centered

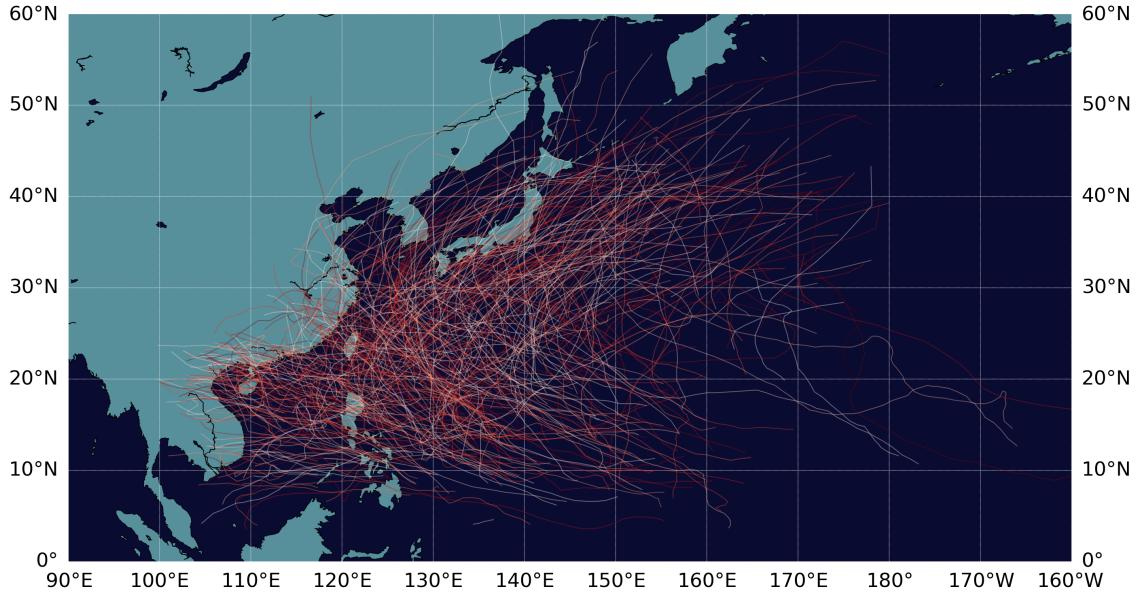


Figure 5: Visualization of typhoon trajectories from the CMA dataset.

on the typhoon eye is selected to capture the storm’s environmental context. The RCAB module then refines the input by selecting the most relevant isobaric planes for each typhoon event.

The model is trained using the Adam optimizer, with Mean Absolute Error (MAE) as the loss function. The Great Circle Distance metric is employed to measure the accuracy of trajectory predictions, computed as:

$$D = R \cos^{-1} [\sin(\phi_1) \sin(\phi_2) + \cos(\phi_1) \cos(\phi_2) \cos(\lambda_2 - \lambda_1)], \quad (16)$$

where ϕ_1 , λ_1 and ϕ_2 , λ_2 are the latitude and longitude coordinates of the predicted and actual typhoon locations, respectively, and R is the Earth’s radius.

Training and Testing dataset. The AM-ConvGRU model was trained and evaluated on typhoon trajectory data spanning from 2000 to 2018, with typhoons from 2015 to 2018 used for testing. By leveraging both 2D meteorological features and 3D atmospheric data, AM-ConvGRU was able to reduce trajectory prediction errors in comparison to existing deep learning models like ConvLSTM. The introduction of the RCAB module proved critical in improving feature selection, particularly at different isobaric pressure levels, leading to better trajectory estimates. Moreover, due to the use of ConvGRU instead of ConvLSTM, the model achieved lower computational complexity and reduced overfitting, while maintaining high predictive performance. The Wide & Deep framework further enhanced generalization by effectively combining statistical learning with spatio-temporal deep learning models.

4 QUANTUM COMPUTING INTEGRATION & FEASIBILITY

As we have detailed the QT and QPA methods, along with their connection to classical ML models in Sec. 2, this section first delves deeper into the theoretical foundations of the QT method, which serves as the basis for QPA. Next, given that our current formulations rely on gate-based quantum circuits, we explore their compatibility with Pasqal’s neutral atom devices and investigate potential implementations of QT and QPA on such hardware. Finally, we discuss the energy efficiency of the proposed method.

4.1 THEORETICAL PERSPECTIVE OF QUANTUM-TRAIN-BASED METHODS

As we have discussed the computational flow of QT and QPA in Sec. 2, it is essential to conduct a theoretical analysis of the error performance of this class of methods. This analysis aims to examine their behavior and the scaling of hyperparameters. Given the complexity of the theoretical discussion, the notation in this

section may differ slightly from that in previous sections; however, all notations consistently describe the same underlying concepts.

4.1.1 QUANTUM-TRAIN RECAP

Start from the beginning again, given a classical NN model with weight $W \in \mathbb{R}^m$, QT framework utilizes a quantum system with $n_{qt} = \lceil \log_2 m \rceil$ qubits, with the Hilbert space size $2^{\lceil \log_2 m \rceil} \geq m$, intend to generate W from the quantum state. Presented as a parameterized quantum circuit operates in an initial state, $|\psi(\theta_{\text{qnn}})\rangle = U(\theta_{\text{qnn}})|0\rangle^{\otimes n_{qt}}$. The measurement probabilities of the first m basis in the quantum state can be formed as a vector $\Psi \in [0, 1]^m$,

$$\Psi = [|\langle \phi_1 | \psi \rangle|^2, |\langle \phi_2 | \psi \rangle|^2, \dots, |\langle \phi_m | \psi \rangle|^2], \quad (17)$$

where $|\phi_i\rangle \in \{0, 1\}^{n_{qt}}$ is i -th the basis state, and $|\langle \phi_i | \psi \rangle|^2 \in [0, 1]$ is the corresponding measurement probability. The first m basis states can also form a matrix $\Phi \in \{0, 1\}^{m \times n_{qt}}$

$$\Phi = [|\phi_1\rangle, |\phi_2\rangle, \dots, |\phi_m\rangle]. \quad (18)$$

A matrix $\Psi^c \in [0, 1]^{m \times (n_{qt}+1)}$ representing the basis and its measured probability are then constructed by:

$$\Psi^c = \begin{bmatrix} |\phi_1\rangle & \parallel & |\langle \phi_1 | \psi \rangle|^2 \\ |\phi_2\rangle & \parallel & |\langle \phi_2 | \psi \rangle|^2 \\ \vdots & & \\ |\phi_m\rangle & \parallel & |\langle \phi_m | \psi \rangle|^2 \end{bmatrix}, \quad (19)$$

where \parallel is the concatenation operation, this means:

$$\Psi_1^c = |\phi_1\rangle \parallel |\langle \phi_1 | \psi \rangle|^2 \quad (20)$$

$$= \underbrace{[0, 0, \dots, 0]}_{n_{qt} \text{ elements}}, |\langle \phi_1 | \psi \rangle|^2. \quad (21)$$

Since $|\langle \phi_i | \psi \rangle|^2 \in [0, 1]$ and $W_i \in \mathbb{R}$ in general, to relate $|\langle \phi_i | \psi \rangle|^2$ and W_i , it is necessary to use a mapping model $G_{\theta_{\text{mm}}}$, parameterized by θ_{mm} to map $|\langle \phi_i | \psi \rangle|^2$ from $[0, 1]$ to \mathbb{R} . The mapping model $G_{\theta_{\text{mm}}}$ is constructed in the form of multilayer perceptron (MLP) with input size $n_{qt} + 1$, such that m could be considered as the batch size while performing the map, and the number of parameters of $G_{\theta_{\text{mm}}}$ can be easily controlled within $O(\text{poly}(n_{qt}))$. Consider the 2-layer case that

$$G_{\theta_{\text{mm}}}(\Psi_i^c) = g_2 \circ \sigma(g_1(\Psi_i^c)) = W_i, \quad (22)$$

where g_1 and g_2 are linear layer operations represented by the combination of matrix multiplication and vector addition, and W_i is the i -th elements of the classical NN weight W , sigmoid function σ is utilized to introduce the non-linearity, and \circ is used to denote the composition of functions. Recalling the dependency of Ψ^c on θ_{qnn} , it can be observed that the classical NN weight $W = W(\theta_{\text{qnn}}, \theta_{\text{mm}})$, that is, tuning the QT parameters θ_{qnn} and θ_{mm} effectively adjusts the target NN weight W .

4.1.2 ERROR PERFORMANCE ANALYSIS

In the following analysis, we adopt the method introduced in (Qi et al., 2023) and extend its application to the QT framework. Assume the target NN with weight $W \in \mathbb{R}^m$ is formed with matrix $w \in \mathbb{R}^{l \times k}$ and bias $b \in \mathbb{R}^k$, such that $m = (k+1)l$, with sigmoid activation function. Given an input data $\mathbf{x} \in \mathbb{R}^l$, the prediction y for this model is

$$y(\mathbf{x}) = \sigma(w^T \mathbf{x} + b). \quad (23)$$

Define \mathbb{F}_{NN} as the classical NN hypothesis space, which represents a collection of NN operators. Given a set of N training data drawn independent and identically distributed from a data distribution \mathcal{D} , for a loss function ℓ and the smooth target function $h_{\mathcal{D}}^*$, defining the optimal NN operator $f_{\mathcal{D}}^* \in \mathbb{F}_{\text{NN}}$, the expected loss can be defined as:

$$\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\ell(h_{\mathcal{D}}^*(\mathbf{x}), f_{\mathcal{D}}^*(\mathbf{x}))], \quad (24)$$

and the empirical loss to minimize $\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)$ is defined as:

$$\mathcal{L}_{\mathcal{S}}(f_{\mathcal{D}}^*) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \ell(h_{\mathcal{D}}^*(\mathbf{x}_n), f_{\mathcal{D}}^*(\mathbf{x}_n)). \quad (25)$$

Now we further define f_D^* the optimal NN operator, f_S^* the empirical optimal operator, \bar{f}_S the returned NN operator. Without QT, using the error decomposition technique, the expected loss $\mathcal{L}_{\mathcal{D}}(\bar{f}_S)$ is factorized into:

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}(\bar{f}_S) &= \underbrace{\mathcal{L}_{\mathcal{D}}(f_D^*)}_{\text{Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^*) - \mathcal{L}_{\mathcal{D}}(f_D^*)}_{\text{Estimation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f_S^*)}_{\text{Training Error}} \\ &\leq \mathcal{L}_{\mathcal{D}}(f_D^*) + 2\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}}) + v \end{aligned} \quad (26)$$

where $\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}})$ is the empirical Rademacher complexity over the family \mathbb{F}_{NN} . To introduce the effect of QT, we define the \mathbb{F}_{QT} as the QT hypothesis space where $\mathbb{F}_{\text{QT}} \subset \mathbb{F}_{\text{NN}}$. Recall that \bar{f}_S is the returned NN hypothesis based on the training dataset S , in the case of QT, we define the \bar{f}_S^{QT} as the returned QT hypothesis while using QT, and $f_S^{\text{QT}*}$ the optimal QT operator. The QT approximation error is defined as $\mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*)$, this defines the error of how the optimal QT generated NN approximates the general target NN with arbitrary weight. While training process in QT involves only the update of θ_{qnn} and θ_{mm} , the training error is defined as $\mathcal{L}_{\mathcal{D}}(\bar{f}_S^{\text{QT}}) - \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*})$, indicating that the training is the process from \bar{f}_S^{QT} toward its optimal $f_S^{\text{QT}*}$.

In this sense, consider the expected loss $\mathcal{L}(\bar{f}_S^{\text{QT}})$ while using QT, the error decomposition technique (Fig. 15) reads:

$$\begin{aligned} \mathcal{L}(\bar{f}_S^{\text{QT}}) &= \underbrace{\mathcal{L}_{\mathcal{D}}(f_D^*)}_{\text{Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^*) - \mathcal{L}_{\mathcal{D}}(f_D^*)}_{\text{Estimation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*)}_{\text{QT Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(\bar{f}_S^{\text{QT}}) - \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*})}_{\text{QT Training Error}} \\ &\leq \mathcal{L}_{\mathcal{D}}(f_D^*) + 2\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}}) + \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) + v \end{aligned} \quad (27)$$

while the bounds for approximation error and estimation error for classical NN should be known, and v refers to the training error that results from the optimization bias of gradient-based algorithms. We focus on the QT approximation error in the following.

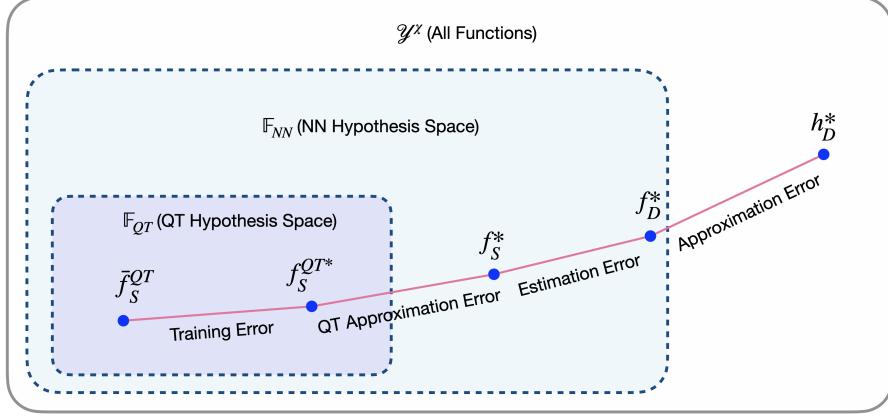


Figure 6: Error decomposition technique with QT.

With the derivation detail in the Appendix A, it can be shown that using mean value theorem, Hoeffding's inequality, Solovay–Kitaev theorem (Dawson & Nielsen, 2005), and the universal approximation theorem (Barron, 1993), the QT approximation error reads:

$$\begin{aligned} &\mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) \\ &\leq \frac{1}{4\sqrt{h_g}} \|\mathbf{1}\mathbf{x}\|_1 + \frac{1}{4} \|b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \quad (\text{Universal Approx.}) \\ &\leq \frac{1}{4} \|b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{h_g}}\right) + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\ &\leq \mathcal{O}\left(\frac{1}{\sqrt{h_g}}\right) + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right). \end{aligned} \quad (28)$$

Eq. 28 establishes an upper bound on the QT approximation error, incorporating key terms that contribute to the overall error performance of the QT framework. Specifically, it provides insights into how different

factors—such as the width of the classical mapping model h_g , the number of quantum measurement shots M , and the depth of the QNN L —affect the error. It is worth noting that in our context, a smaller c_{sk} means that fewer gates are required to approximate the given target unitary. Thus, it is natural to consider that c_{sk} is determined by the QNN architecture and can be reduced through QNN architecture search. Here, we use the behavior of this approximation to present the precision ϵ decreases exponentially with the sequence length $L^{1/c_{sk}}$.

4.2 COMPATIBILITY

Although the framework described above may initially appear to be tailored for gate-based quantum computers, according to the Pasqal’s official publication (Henriet et al., 2020), neutral atom QPUs are able to implement both digital and analog quantum processing task. To investigate the compatibility and feasibility of our method to the current **Pasqal’s neutral atom technology**, we should emphasize that the QT/QPA concept is applicable to **any** parameterizable quantum computing architecture, with possible extension to digital-analog quantum computing paradigm.

4.2.1 DIGITAL QUANTUM COMPUTING

The feasibility of digital quantum computing on Pasqal’s system primarily arises from its ability to implement one- and two-qubit gate operations using highly controlled laser pulses. Single-qubit gates are realized through Rabi oscillations, which enable arbitrary rotations on the Bloch sphere with fidelities exceeding 99.5%. More crucially, two-qubit entangling gates (CNOT gates with about 94.1% fidelities) are implemented via Rydberg blockade interactions, an effect that prevents two nearby atoms from being excited to certain quantum states simultaneously. This mechanism allows for the execution of controlled two-qubit gates, including the Controlled-NOT (CNOT) gate, a fundamental component of universal digital quantum computation.

The presence of a universal gate set, composed of arbitrary single-qubit rotations and CNOT gates, ensures that any quantum algorithm, including quantum neural networks (QNNs), can be theoretically executed on Pasqal’s system. Furthermore, Pasqal’s high connectivity architecture facilitates more efficient gate implementation compared to superconducting qubit systems, where swap operations are often required to move quantum information across a fixed lattice structure. The scalability of neutral atom processors, demonstrated in experiments with 100 to 1,000 qubits, suggests that Pasqal’s platform could support mid-scale quantum circuits without excessive qubit overhead. In principle, these capabilities indicate that a fully digital QNN could be implemented on Pasqal’s hardware, provided the required number of gate operations remains within feasible limits given current hardware constraints.

However, while the theoretical feasibility of digital quantum computing on Pasqal’s platform is well established, its practical implementation remains an evolving challenge. Current experimental limitations, including gate fidelities, decoherence times, and computational overhead, introduce constraints that impact the viability of deep digital quantum circuits.

4.2.2 DIGITAL-ANALOG QUANTUM COMPUTATIONAL PARADIGM

A hybrid digital-analog approach provides a more efficient and hardware-optimized alternative by leveraging the strengths of both computation models. Here we show some possible approach to extend our method to the digital-analog setting. Since QNNs rely on multiple layers of unitary transformations, deep gate-based implementations suffer from rapid degradation of computational accuracy. In contrast, analog quantum computation, which directly evolves quantum states under engineered Hamiltonians, allows for a more efficient and noise-resistant execution of quantum algorithms. By integrating analog evolution into QNN architectures, the need for long digital gate sequences can be minimized, thus mitigating the impact of gate errors and improving overall computational stability.

As we have mentioned that the QT concept is applicable to **any** parameterizable quantum computing architecture, including Pasqal’s neutral atom technology. The key adaptation depends on the specific hardware scenario—one may need to parameterize the Hamiltonian accordingly. For instance, in the case of neutral atom platforms, the Hamiltonian can be expressed as:

$$H = \sum_{k=1}^N (\cos(\phi)\sigma_k^x - \sin(\phi)\sigma_k^y) - \frac{1}{2}\delta(t) \sum_{k=1}^N (\mathbb{I}_k - \sigma_k^z) + \sum_{i < j}^N \frac{C_6}{R_{ij}^6} \hat{n}_i \hat{n}_j. \quad (29)$$

While the Hamiltonian could be parameterized, the evolution of quantum state could be viewed as the parameterized unitary operation

$$|\psi(t)\rangle = \exp\left(-i \int_0^t H(\tau) d\tau\right) |\psi(0)\rangle \quad (30)$$

This formulation allows for the realization of a similar parameterized learning process as in the gate-based scenario, demonstrating the flexibility of QT across different quantum computing paradigms. By modulating the evolution time and interaction strength, the resulting dynamics can be used to encode unitary transformations without additional gate operations. Moreover, a key requirement in training QNNs is the ability to optimize parameters efficiently. In a purely digital model, variational training relies on iterative gradient-based updates, requiring multiple circuit evaluations for each parameter step

$$\frac{\partial \langle \psi(\theta) | H | \psi(\theta) \rangle}{\partial \theta} = \frac{\langle \psi(\theta + \epsilon) | H | \psi(\theta + \epsilon) \rangle - \langle \psi(\theta - \epsilon) | H | \psi(\theta - \epsilon) \rangle}{2\epsilon}, \quad (31)$$

where H is the loss function Hamiltonian and ϵ is a small perturbation. This method requires repeated execution of quantum circuits, increasing the computational cost. In a hybrid digital-analog setting, parameter updates can be embedded into Hamiltonian evolution using time-dependent control

$$H(\theta, t) = H_0 + \sum_i \theta_i(t) H_i, \quad (32)$$

where the parameters $\theta_i(t)$ evolve continuously instead of being updated in discrete steps. This approach reduces the need for multiple circuit evaluations, leading to faster convergence with lower computational overhead.

Connectivity Advantages in Neutral Atom QNNs. Pasqal's neutral atom qubits feature reconfigurable connectivity, allowing qubits to be arranged dynamically during computation. Unlike superconducting qubits, where long-range interactions require swap gates, neutral atoms can be moved in real-time within optical tweezers, allowing for direct interaction between distant qubits. This capability reduces the number of required entangling operations, enhancing the efficiency of QNN layers. In a digital QNN, each layer would require multiple two-qubit gates to propagate quantum information across the network. Using a hybrid approach, multi-qubit interactions can be realized naturally via collective Rydberg blockade interactions, described by the Hamiltonian

$$H_{\text{collective}} = \sum_{i < j} \Omega_{ij} |r_i r_j\rangle \langle r_i r_j| + \Delta |r_i\rangle \langle r_i|, \quad (33)$$

where Ω_{ij} represents the effective coupling strength between atoms. This collective interaction allows multiple weight transformations to be applied simultaneously, reducing the total number of computational steps required.

The incorporation of continuous parameter updates and reconfigurable connectivity further enhances computational efficiency, making hybrid QNNs more viable than purely digital models in near-term quantum hardware. As Pasqal progresses toward fully digital quantum computing with logical qubits, hybrid models will play a critical role in bridging the gap between near-term feasibility and long-term scalability, ensuring that quantum neural networks remain adaptable to the evolving capabilities of neutral atom quantum processors.

4.3 ENERGY EFFICIENCY

4.3.1 VIA PARAMETER REDUCTION IN QUANTUM-TRAIN

In classical ML, the training cost is directly proportional to the number of parameters being optimized. A neural network with millions or billions of parameters requires high-power GPU clusters, leading to substantial energy consumption. The cost function for training a classical deep learning model can be expressed as

$$C_{\text{classical}} = O(P \cdot D), \quad (34)$$

where P is the number of trainable parameters, and D represents the number of training iterations. Large-scale models, such as transformers or deep neural networks, can reach parameter counts in the hundreds of billions, making energy consumption a growing concern in AI research. With QT significantly reduces the

number of parameters to polylogarithmic scaling as described in Sec. 2, the number of trainable parameters is

$$P_{\text{quantum}} = O(\text{polylog}(P)). \quad (35)$$

Since training time complexity often scales at least linearly with parameter count, this means that the overall computational workload is reduced to:

$$C_{\text{quantum}} = O(\text{polylog}(P) \cdot D_{\text{quantum}}), \quad (36)$$

where D_{quantum} represents the quantum-enhanced training iterations. This leads to a drastic reduction in the number of floating-point operations (FLOPs) required for training, directly translating to lower energy consumption.

4.3.2 IN THE CONTEXT OF PASQAL’S NEUTRAL ATOM QUANTUM COMPUTING

Pasqal’s neutral atom quantum processors do not require dilution refrigerators, unlike superconducting qubits, and operate at room temperature, significantly reducing cooling-related energy costs. While the paper does not explicitly discuss power consumption, it highlights the high coherent evolution rate of Rydberg atom qubits, which suggests that quantum operations are performed with a lower energy overhead per computation.

Given that classical deep learning models consume massive energy resources, an approach that reduces model size while maintaining performance directly contributes to quantum energy efficiency.

By leveraging analog Hamiltonian evolution for parameterized transformations in Pasqal’s hybrid quantum system, further reductions in quantum gate overhead can be achieved. The energy cost of digital gate execution can be approximated as

$$E_{\text{digital}} = N_{\text{gates}} \cdot E_{\text{gate}}, \quad (37)$$

where N_{gates} is the number of two-qubit entangling gates executed, and E_{gate} represents the energy per gate operation. If a hybrid quantum-analog approach reduces the required number of digital gates, then total energy consumption decreases:

$$E_{\text{hybrid}} \approx \alpha E_{\text{digital}}, \quad 0 < \alpha < 1, \quad (38)$$

suggesting an energy efficiency gain proportional to the reduction in digital gate execution.

Our QT/QPA approaches mentioned in Sec. 2 aligns strongly with energy efficiency principles, as it reduces the number of parameters required for model training in a polylogarithmic scale, leading to a drastic reduction in training cost and computation overhead. Since Pasqal’s neutral atom quantum processors operate with lower energy demands than superconducting qubits and offer analog computation advantages, implementing our method in this setting could further enhance efficiency. By leveraging a hybrid digital-analog computation model, energy-intensive digital gate execution can be minimized, leading to better resource utilization and lower power consumption per computational step. Given the exponential increase in energy consumption for training state-of-the-art classical ML models, a quantum-enhanced training strategy that significantly reduces parameter complexity offers a scalable, energy-efficient alternative, making it a strong candidate for real-world AI applications.

5 IMPLEMENTATION, CODE & TESTING

The implementation of this project is primarily built using Pasqal’s Qadence ([Seitz et al., 2025](#)) and PyTorch. The GitHub repository for this project can be accessed at (https://github.com/CYLphysics/QPA_Typhoon_Trajectory).

5.1 PROJECT ARCHITECTURE

The file architecture is illustrated in Fig. 7. The `code_base` directory contains all the necessary source files required to implement the results presented in this report. The `test` folder provides Jupyter notebooks demonstrating how to reproduce these results using the code from `code_base`. Additionally, the predicted trajectories and baseline trajectory data are stored in the `QPA_track_data` directory.

5.2 MWE & TESTING

In the `test` folder, we provide several Jupyter notebooks for minimal working examples (MWE) and testing.

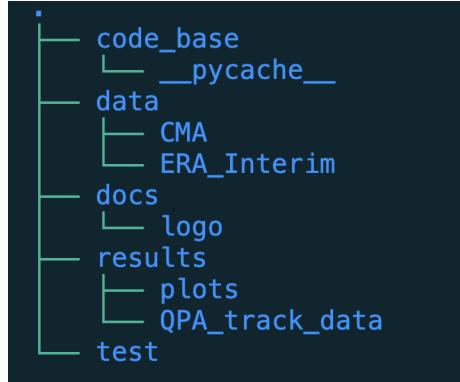


Figure 7: The file tree of this project.

1. `QPA_ck_64_qnn_depth_20.ipynb`: In this notebook, we demonstrate the end-to-end training and testing workflow for applying Quantum Parameter Adaptation (QPA)—a quantum-enhanced method—to efficiently train the classical ConvGRU model using only 2.57% of the original trainable parameters, while maintaining competitive performance.
2. `Classical_Model_baseline.ipynb`: This notebook presents the pure classical training baseline for the original ConvGRU model, which consists of 8 million trainable parameters.
3. `Classical_Model_pruning_01.ipynb` and `Classical_Model_weight_sharing_1.ipynb`: Since QPA is essentially a quantum-enhanced parameter-efficient learning method, we also compare it with two well-known classical neural compression techniques: pruning and weight sharing. The comparison results are recorded and visualized in `Plot_notebook_quantum_vs_classical_result.plot.ipynb`.

6 RESULTS, EVALUATION & ROADMAP

In this section, we present the results obtained from QPA and compare them with classical model compression methods, such as pruning and weight sharing. Additionally, we analyze the performance of QPA under different hyperparameter settings, including varying chunk sizes per basis n_{mlp} and different depths of QNN layers L .

6.1 PERFORMANCE & BENCHMARKS

6.1.1 OVERALL BENCHMARKING

The QPA setting in the following experiments is as follows: within the ConvGRU model for typhoon trajectory forecasting, QPA is applied to the last two linear layers, while LoRA is applied to the remaining layers. The classical full model consists of 8,399,540 (8.39M) trainable parameters. In contrast, when QPA is used with different configurations, as shown in Fig. 8, the number of trainable parameters is reduced to approximately 0.2M to 0.3M. Since QPA is fundamentally a parameter-efficient learning method that compresses the number of trainable parameters, it is crucial to compare it with other classical compression techniques, such as pruning (Neill, 2020; Blalock et al., 2020) and weight sharing (Nowlan & Hinton, 2018).

Fig. 8 illustrates the performance comparison of different model compression methods applied to typhoon trajectory forecasting on a testing dataset spanning 2015 to 2018. The x-axis represents the number of trainable parameters, while the y-axis denotes the total average error (km) for the forecasting task, where lower values indicate higher forecasting accuracy. The objective of this study is to assess the effectiveness of various compression techniques, including QPA, weight sharing, and pruning, in reducing model size while maintaining predictive accuracy.

The classical full model, as proposed in the original paper (Xu et al., 2022), is represented by an orange star and achieves the lowest forecasting error but requires significantly more trainable parameters, exceeding 8.39M. While larger models tend to achieve higher accuracy, their computational and energy costs can be prohibitive. Pruning, represented by red circles, demonstrates a steep decline in error as the number of

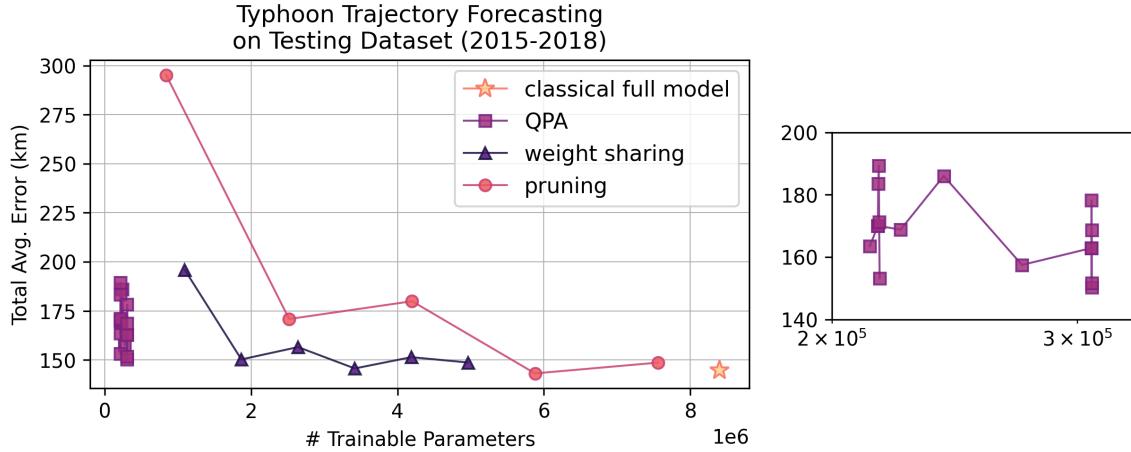


Figure 8: Typhoon trajectory forecasting on testing dataset (2015-2018).

trainable parameters decreases. However, after reaching approximately 10^6 parameters, the reduction in error plateaus, suggesting diminishing returns from further pruning.

The weight sharing method, depicted by dark blue triangles, offers a more balanced trade-off between parameter reduction and forecasting accuracy. Compared to pruning, weight sharing maintains relatively stable accuracy with significantly fewer trainable parameters, indicating that it is an effective strategy for improving model efficiency. Meanwhile, QPA, shown as purple squares, consistently achieves competitive performance with a significantly lower parameter count. The clustering of QPA results on the left side of the plot within the 0.2M to 0.3M parameter range highlights its potential to maintain accuracy while substantially reducing computational requirements. The inset plot provides a magnified view of QPA's performance, specifically within the 0.2M to 0.3M parameter range. While minor fluctuations in total average error are observed, the results remain within a narrow band, reinforcing QPA's robustness even under significant parameter compression. This suggests that QPA provides a viable alternative to traditional compression methods by leveraging quantum-inspired techniques to optimize model efficiency. The content of this inset plot is expanded in Fig. 9. The results suggest that QPA, a quantum method leveraging quantum principles, offers a promising alternative to classical full models by representing high-dimensional complex data with fewer trainable parameters. By utilizing quantum state representations mappings, QPA efficiently captures intricate data structures, enabling substantial reductions in model size while preserving predictive accuracy. While pruning proves effective at higher parameter levels, it reaches a performance plateau beyond a certain threshold, limiting its scalability. In contrast, QPA's ability to compress information using quantum techniques allows it to maintain competitive forecasting accuracy while significantly lowering computational complexity. These findings underscore the potential of QPA as an energy-efficient quantum-enhanced approach for machine learning applications, offering a scalable and resource-efficient solution for real-world forecasting tasks.

6.1.2 ON HYPERPARAMETERS OF QPA

Fig. 9 consists of three subplots, each examining the impact of different QPA hyperparameters on forecasting accuracy. The general trend observed across all subplots suggests that increasing the number of trainable parameters tends to improve forecasting performance, resulting in lower error values. Fig. 9(a), the QNN layer depth is fixed at $L = 4$, while the chunk size n_{mlp} is varied across the values 32, 64, 128, 256, 512, and 768. The results indicate that the total average error fluctuates across different chunk sizes, but there is no strictly monotonic trend. However, configurations with higher trainable parameter counts generally achieve lower errors, reinforcing the parameter-performance trade-off in quantum models.

In Fig. 9(b), the chunk size is fixed at $n_{mlp} = 64$, while the number of QNN layers is varied over $L = 4, 8, 12, 16, 20$. The results exhibit a similar trend, where increasing QNN depth initially leads to higher error values, possibly due to optimization difficulties in shallow models. However, as the number of trainable parameters increases, the total average error decreases, suggesting that a deeper QNN architecture improves prediction accuracy. Fig. 9 (c), the experiment is identical to (b) but with a larger chunk size fixed at

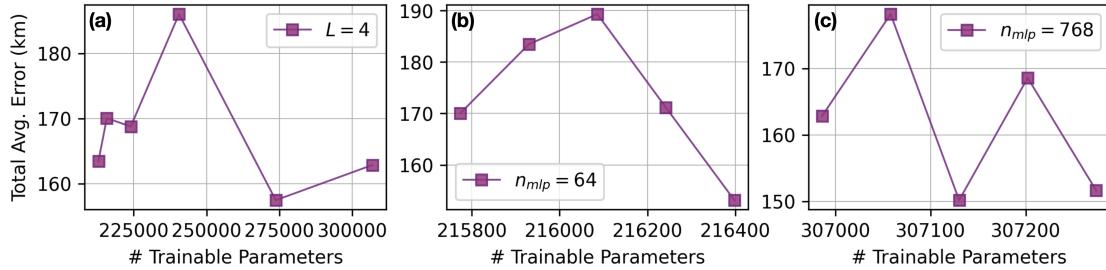


Figure 9: Typhoon trajectory forecasting on testing dataset (2015-2018). (a) Fixing # of QNN layers $L = 4$, the corresponding required qubit size from left to right is 9, 8, 7, 6, 5, 4. (b) and (c): fixing n_{mlp} and varying the QNN depth $L = 4, 8, 12, 16, 20$.

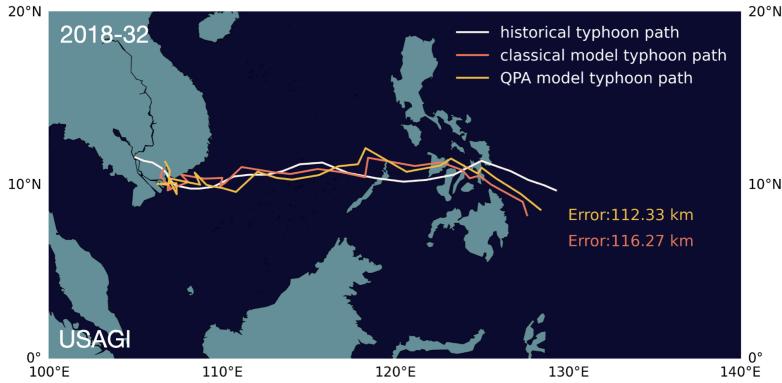


Figure 10: Typhoon trajectory forecasting results for typhoon USAGI, # 2018-32.

$n_{mlp} = 768$. A similar pattern emerges, where models with more trainable parameters achieve lower errors, reinforcing the idea that larger models can capture more complex features and improve generalization.

6.1.3 RESULTING TRAJECTORIES VISUALIZATION

To better understand the performance of QPA, it is beneficial to visualize the predicted typhoon trajectories. The ability to accurately forecast typhoon paths is crucial for early warning systems, as it enables timely evacuations, reduces financial losses, and, most importantly, **saves lives**. The effectiveness of a forecasting model is not only measured by numerical accuracy but also by how well it captures the essential patterns and deviations in real-world trajectories.

In Fig. 10 and Fig. 11, we present two cases where QPA, with approximately 0.3M trainable parameters, outperforms the original model with 8.3M trainable parameters. These results demonstrate that QPA, despite using only a fraction of the trainable parameters, effectively captures complex atmospheric dynamics and provides trajectory forecasts that closely align with the actual typhoon paths. It is particularly impressive that even with a significantly reduced parameter count—nearly dozens of times smaller than the full model—the deviation in error remains minimal. This suggests that QPA’s quantum-inspired parameterization efficiently encodes critical weather patterns, making it a promising approach for resource-efficient and scalable typhoon prediction. Conversely, Fig. 12 and Fig. 13 illustrate cases where QPA does not surpass the original full model. While QPA still provides reasonably accurate predictions, these results highlight the limitations of parameter-efficient models in capturing extreme meteorological variations, which may require larger network capacity to fully resolve. Nevertheless, the fact that QPA maintains competitive performance with significantly fewer parameters suggests that it is a viable approach for improving forecasting efficiency. These instances indicate that while QPA is an effective method for reducing computational costs, further

refinements—such as hybrid quantum-classical architectures or adaptive parameterization techniques—may enhance its performance in more complex forecasting scenarios.

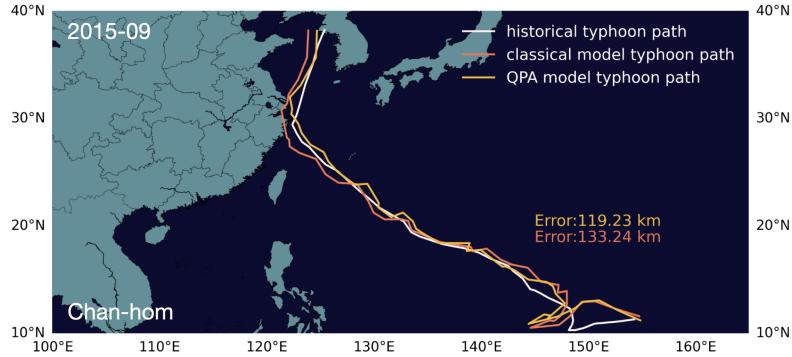


Figure 11: Typhoon trajectory forecasting results for typhoon Chan-hom, # 2015-09.

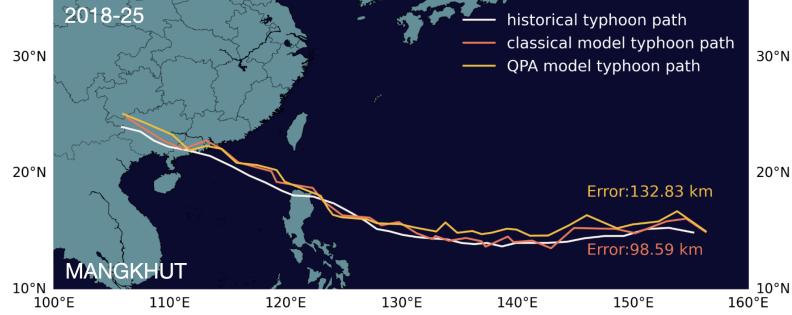


Figure 12: Typhoon trajectory forecasting results for typhoon MANGKHUT, # 2018-25.

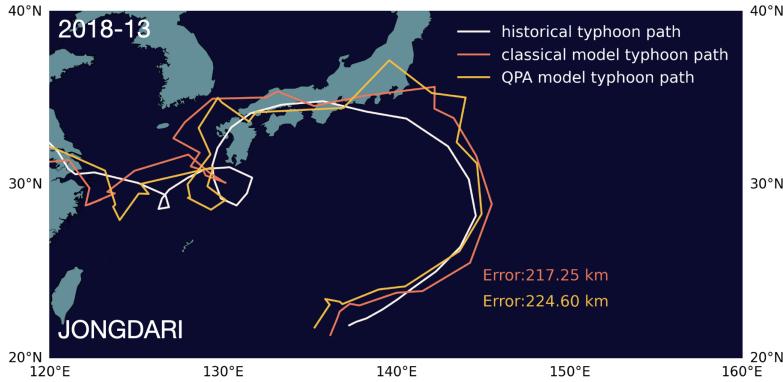


Figure 13: Typhoon trajectory forecasting results for typhoon JONGDARI, # 2018-13.

6.2 DEPLOYMENT CHALLENGES

The deployment of quantum-enhanced parameter-efficient learning models, such as QT and QPA introduced in this work for typhoon trajectory forecasting case, presents an exciting opportunity for advancing real-world applications. By significantly reducing the number of trainable parameters while maintaining predictive accuracy, these methods offer a promising direction for more efficient and scalable machine learning models. Successfully implementing QPA on Pasqal’s neutral atom quantum computing platform will require targeted

optimizations to fully harness its potential, paving the way for future breakthroughs in quantum machine learning.

6.2.1 COMPATIBILITY

As already detailed in Sec. 4.2, a key aspect of deployment is ensuring compatibility between QPA and Pasqal’s neutral atom quantum processors. Given that Pasqal’s hardware supports both digital and digital-analog quantum computation, there is a clear pathway for integrating QPA into near-term quantum systems. As neutral atom devices leverage Rydberg interactions for controlled entanglement, further optimizations in circuit depth and connectivity will enhance performance and reliability. Additionally, ongoing research in error mitigation strategies and hardware-aware optimizations will further refine QPA-based learning, ensuring it remains robust against quantum noise and decoherence effects. Integrating QPA into existing high-performance computing (HPC) environments represents another exciting frontier. Most large-scale weather forecasting models, including typhoon trajectory prediction systems, are designed for classical supercomputing infrastructure. The transition to a hybrid quantum-classical framework will unlock new opportunities for faster, more energy-efficient computations, provided that efficient interfacing techniques are developed. By optimizing data flow and parallel execution between quantum and classical processors, large-scale deployments can seamlessly leverage quantum advantages while maintaining compatibility with existing computational pipelines.

6.2.2 CHALLENGE OF MEASUREMENT

Another important consideration is the efficient handling of quantum measurements within QPA. Unlike classical deep learning models, which utilize backpropagation for efficient parameter updates, QPA relies on quantum circuit evaluations and measurement-based updates. While this introduces additional measurement costs, advancements in quantum measurement compression techniques offer promising solutions to reduce overhead while maintaining learning accuracy. A proposed method called Generative Interpolation in the ICLR 2025 workshop paper (Liu et al., 2025a) highlights similar challenges and suggests practical strategies for optimizing quantum measurement efficiency. With continued improvements in quantum state extraction and hybrid quantum-classical learning, the measurement cost barrier can be significantly reduced, making large-scale quantum training more feasible. Scalability remains a fundamental goal in the advancement of QPA-based forecasting models. While QPA has demonstrated the ability to compress model parameters to a polylogarithmic scale, real-world forecasting tasks require handling vast amounts of meteorological data across different time horizons. Ensuring that QPA-generated parameters remain stable and adaptable across diverse atmospheric conditions will be crucial for achieving reliable long-term forecasts. Future work in adaptive quantum parameter tuning, where quantum-enhanced fine-tuning dynamically adjusts based on real-time atmospheric variability, could further enhance the robustness of quantum-assisted forecasting.

QPA presents a novel and transformative approach for energy-efficient, scalable typhoon trajectory forecasting. While continued advancements in quantum hardware, quantum-classical interfacing, and adaptive learning techniques will be necessary, these developments are already progressing rapidly. As quantum computing technology continues to mature, the integration of QPA into operational forecasting systems holds immense potential for improving climate modeling, enhancing disaster preparedness, and ultimately protecting lives and infrastructure from extreme weather events. By addressing current challenges with strategic innovation, QPA is well-positioned to revolutionize the intersection of quantum computing and climate science, opening the door to a new era of high-efficiency, real-world quantum applications.

6.3 CARBON FOOTPRINT ANALYSIS

In this section, we discuss the carbon footprint analysis using the sustainability benchmark provided by Pasqal. Since the use-case emissions (including preparation) in tCO₂ are estimated based on the runtime of the corresponding quantum and classical hardware, we first estimate the runtime of QPA compared to the classical approach under an idealized near-future assumption. Using this estimated runtime, we then derive the corresponding carbon emissions based on the provided sustainability benchmark.

6.3.1 RUNTIME ESTIMATION

The efficiency of machine learning models is heavily influenced by the number of trainable parameters, as training time is typically dependent on the computational complexity of parameter updates. In deep learning, training time T generally scales with the number of parameters P according to the relationship $T \propto P^c$,

where c reflects the complexity of the computational steps involved. In an optimized training pipeline, linear scaling ($c = 1$) represents highly efficient matrix multiplications, while quadratic scaling ($c = 2$) arises in more computationally demanding architectures, such as attention-based models. Although the results presented in this report are derived from simulations using the Qadence simulator from Pasqal, the estimated runtime presented here may not precisely match current simulation outcomes. Instead, these projections represent an idealized scenario, envisioning a future in which quantum computing technology has advanced significantly, leading to optimized performance in real-world applications.

From the experimental results, the classical full model used for typhoon trajectory forecasting consists of 8.3M trainable parameters. In contrast, QPA significantly reduces this requirement to 0.2M parameters, representing a parameter reduction factor of approximately 41.5. Given that training time is closely related to the number of trainable parameters, we estimate the relative runtime of QPA models compared to classical models by analyzing different scaling behaviors.

Assuming linear scaling ($c = 1$), where training time is directly proportional to the number of parameters, the expected training time for QPA is reduced by a factor of 41.5, yielding:

$$T_{\text{QPA}} \approx \frac{T_{\text{classical}}}{41.5} \quad (39)$$

In a more complex scenario where training exhibits quadratic scaling ($c = 2$), the computational cost grows proportionally to the square of the number of parameters, leading to a much larger reduction:

$$T_{\text{QPA}} \approx \frac{T_{\text{classical}}}{1722} \quad (40)$$

Applying these estimates to a classical full model requiring 100 hours of training, a QPA-based model would require approximately 2.4 hours under linear scaling and as little as 3.5 minutes under quadratic scaling. Similarly, for large-scale experiments requiring 1000 GPU-hours, QPA reduces the computational burden to 24.1 GPU-hours in the linear case and just 35 minutes in the quadratic case. These reductions highlight the significant computational savings achieved by QPA, demonstrating its potential to dramatically shorten training times while maintaining competitive forecasting accuracy.

6.3.2 ESTIMATE EMISSION FROM RUNTIME

Based on the previously derived runtime in an idealized near-future scenario, we utilize Pasqal’s provided sustainability benchmark to estimate the tCO₂ emissions. Under these assumptions:

1. QPA is executed on the Orion series, where the qubit requirement is expected to be only in the range of dozens of qubits on a 100–200 qubit device. We assume that the remaining qubits are utilized for efficient quantum error correction to enhance the reliability of the QPA execution.
2. The Post-Orion series, equipped with 400–1000 qubits, provides a 10× parallelization advantage compared to the Orion series. This means that the system can execute 10× more circuits simultaneously, effectively reducing the runtime to 10% of that on the Orion series.
3. The Joliot-Curie Rome supercomputer features over 42 times the number of GPUs compared to a basic GPU server. However, considering both parallelization and the superior GPU architecture, we assume it provides a 100× speedup relative to the basic GPU server.

Using these assumptions, along with Pasqal’s sustainability benchmark, we derive the tCO₂ emissions, as illustrated in Fig. 14.

Fig. 14 presents a comparative analysis of tCO₂ equivalent emissions across different computational platforms for ML model training, considering both highly efficient linear models ($c = 1$) and attention-based models ($c = 2$) described in the previous section. The x-axis represents the model size factor (runtime). The analysis QPA on Pasqal’s Orion and Post-Orion series, alongside classical full-model training on both basic GPU servers and the Joliot-Curie Rome supercomputer.

In the left panel (linear models, $c = 1$), the results indicate that QPA implementations on the Orion (100–200 qubits) and Post-Orion (400–1000 qubits) series exhibit significantly lower emissions compared to the classical method on the Joliot-Curie Rome supercomputer. The emissions from QPA remain relatively stable across different model sizes, demonstrating the efficiency of quantum computing in reducing the carbon footprint of

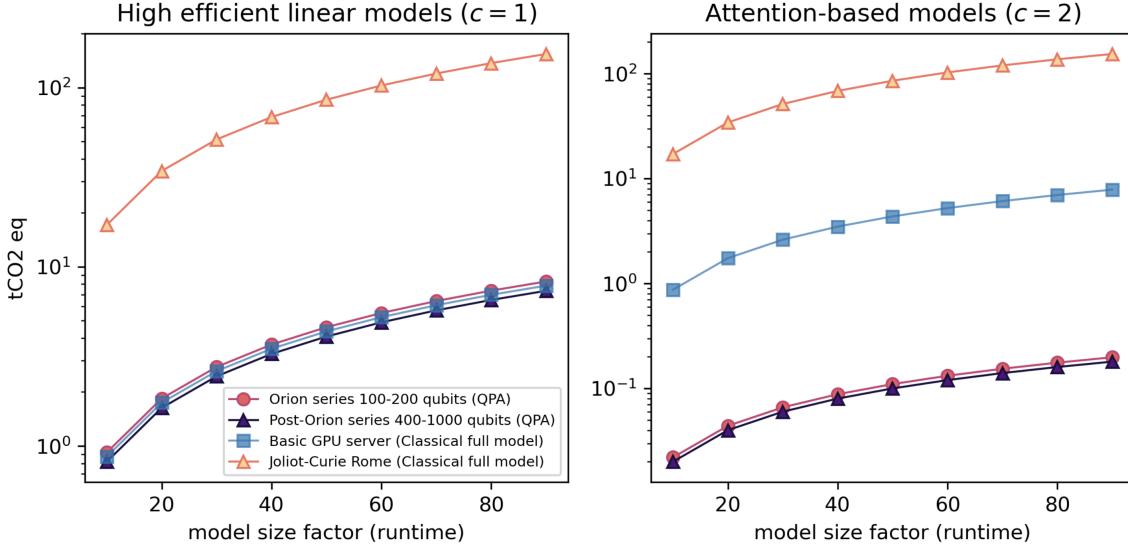


Figure 14: Estimated $t\text{CO}_2$ emissions for QPA on Pasqal’s Orion and Post-Orion series compared to classical full-model training on a basic GPU server and the Joliot-Curie Rome supercomputer, demonstrating the sustainability advantage of quantum-enhanced parameter-efficient learning across both linear ($c = 1$) and attention-based ($c = 2$) models.

parameter-efficient models. Meanwhile, classical full-model training on the basic GPU server shows emissions comparable to those of QPA, which is an interesting observation—suggesting that for sufficiently simple models, a basic GPU server may be sufficient. In the right panel (attention-based models, $c = 2$), where computational demands increase quadratically with model size, the advantages of QPA become even more pronounced. QPA on Orion and Post-Orion maintains low emissions, whereas classical full-model training on the basic GPU server experiences a significantly steeper increase in emissions. The Joliot-Curie Rome supercomputer, despite its computational power, exhibits the highest emissions, reinforcing the environmental cost of large-scale deep learning models trained on traditional HPC infrastructures.

These results highlight the substantial sustainability benefits of QPA-based quantum computing. By efficiently utilizing fewer trainable parameters and leveraging quantum phenomenon, QPA not only reduces computational overhead but also significantly lowers carbon emissions compared to classical full-model training. As quantum hardware continues to evolve, the integration of QPA into large-scale ML workflows has the potential to transform AI sustainability, making high-performance training both computationally and environmentally efficient.

6.4 FUTURE OUTLOOK

As quantum computing technology continues to advance, the integration of QPA into real-world applications presents exciting opportunities for hybrid quantum-classical ML, energy-efficient AI models, and sustainable large-scale computing. While the findings in this report demonstrate the feasibility of QPA on Pasqal’s neutral atom quantum processors, further research and development will be required to unlock its full potential. Future efforts should focus on refining hybrid architectures, improving measurement efficiency, optimizing quantum learning models, and contributing to the sustainability of AI-driven climate forecasting.

6.4.1 LEVERAGING THE POTENTIAL OF NEUTRAL ATOM QUANTUM COMPUTING

A key avenue for future exploration lies in the advancement of hybrid quantum-classical integration. As Pasqal’s neutral atom quantum hardware supports both digital and digital-analog paradigms, QPA could be further optimized by leveraging continuous-time Hamiltonian evolution for learning tasks. By reducing the reliance on gate-based circuits, quantum-native optimization methods could significantly enhance training efficiency. Additionally, integrating hardware-aware error mitigation strategies will be crucial for ensuring

stable and scalable learning on real quantum devices. Future research should focus on developing hybrid training pipelines, where quantum-generated parameter updates interact seamlessly with classical optimization processes, thereby enhancing both model accuracy and computational efficiency.

Another critical challenge to address is the quantum measurement bottleneck, which remains a fundamental limitation in QML implementations. Unlike classical deep learning, where gradients can be efficiently computed through backpropagation, QPA relies on quantum circuit evaluations and measurement-based updates, introducing additional computational overhead. Future work should explore quantum measurement compression techniques, such as optimal quantum-state tomography and hybrid classical-quantum ML-based methods (Liu et al., 2025a), to minimize the number of measurements required while preserving model fidelity. Additionally, circuit evaluation strategies that reduce redundant measurements could further enhance efficiency, making QPA-based models more practical for large-scale applications.

6.4.2 BEYOND CLIMATE FORECASTING

Beyond technical improvements, QPA’s potential role in building a sustainable AI ecosystem is another promising direction. As the carbon footprint analysis suggests, QPA-based learning on Pasqal’s quantum hardware offers a substantial reduction in energy consumption compared to classical full-model training. This efficiency gain could be extended beyond climate forecasting, benefiting resource-intensive AI applications such as large-scale natural language processing (NLP) and financial modeling. Future collaborations between quantum computing researchers, AI practitioners, and sustainability experts will be instrumental in developing quantum-assisted machine learning models that align with global carbon reduction goals.

In the long term, the integration of QPA into next-generation quantum computing platforms will play a pivotal role in shaping the future of hybrid AI systems. As Pasqal moves towards larger, more powerful quantum processors, QPA’s scalability and adaptability will become even more critical. Research efforts should continue to refine quantum-assisted model compression techniques, ensuring that parameter-efficient learning methods remain at the forefront of AI acceleration. By addressing current hardware limitations and improving algorithmic efficiency, QPA has the potential to revolutionize quantum-enhanced AI, unlocking new frontiers in energy-efficient machine learning and real-world AI sustainability.

REFERENCES

- A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. doi: 10.1109/18.256500.
- Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.
- Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.
- Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.
- Sergey Bravyi, Oliver Dial, Jay M Gambetta, Darío Gil, and Zaira Nazario. The future of quantum computing with superconducting qubits. *Journal of Applied Physics*, 132(16), 2022.
- Matthias C Caro, Hsin-Yuan Huang, Marco Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J Coles. Generalization in quantum machine learning from few training data. *Nature communications*, 13(1):4919, 2022.
- Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.
- Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE access*, 8:141007–141024, 2020.
- Christopher M Dawson and Michael A Nielsen. The solovay-kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005.
- Kerry Emanuel. Increasing destructiveness of tropical cyclones over the past 30 years. *Nature*, 436(7051): 686–688, 2005.
- Jay Gambetta. Quantum-centric supercomputing: The next wave of computing. *IBM Research Blog*, 2022.
- Quantum AI Google. Choosing hardware for your qsim simulation. https://quantumai.google/qsim/choose_hw, 2024.
- Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.
- Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022.

- Y.-L. Kuo, Y.-M. Liu, H.-J. Chu, and H.-C. Lee. Are the rich less prone to flooding? a case study on flooding in the southern taiwan during typhoon morakot and typhoon fanapi. *Natural Hazards and Earth System Sciences Discussions*, 2022:1–19, 2022. doi: 10.5194/nhess-2022-38. URL <https://nhess.copernicus.org/preprints/nhess-2022-38/>.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
- Chu-Hsuan Abraham Lin, Chen-Yu Liu, and Kuan-Cheng Chen. Quantum-train long short-term memory: Application on flood prediction problem. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, pp. 268–273, 2024. doi: 10.1109/QCE60285.2024.10290.
- Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020.
- Chen-Yu Liu and Samuel Yen-Chi Chen. Federated quantum-train with batched parameter generation. In *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1133–1138, 2024. doi: 10.1109/ICTC62082.2024.10827567.
- Chen-Yu Liu, En-Jui Kuo, Chu-Hsuan Abraham Lin, Sean Chen, Jason Gemsun Young, Yeong-Jar Chang, and Min-Hsiu Hsieh. Training classical neural networks by quantum machine learning. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, pp. 34–38, 2024a. doi: 10.1109/QCE60285.2024.10248.
- Chen-Yu Liu, En-Jui Kuo, Chu-Hsuan Abraham Lin, Jason Gemsun Young, Yeong-Jar Chang, Min-Hsiu Hsieh, and Hsi-Sheng Goan. Quantum-train: Rethinking hybrid quantum-classical machine learning in the model compression perspective. *arXiv preprint arXiv:2405.11304*, 2024b.
- Chen-Yu Liu, Chu-Hsuan Abraham Lin, Chao-Han Huck Yang, Kuan-Cheng Chen, and Min-Hsiu Hsieh. Qtrl: Toward practical quantum reinforcement learning via quantum-train. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, pp. 317–322, 2024c. doi: 10.1109/QCE60285.2024.10299.
- Chen-Yu Liu, Kuan-Cheng Chen, Samuel Yen-Chi Chen, Huang wei hao, Wei-Jia Huang, and Yen Jui Chang. Frame generation in hilbert space: Generative interpolation of measurement data for quantum parameter adaptation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025a. URL <https://openreview.net/forum?id=JNhTrqu39R>.
- Chen-Yu Liu, Chao-Han Huck Yang, Hsi-Sheng Goan, and Min-Hsiu Hsieh. A quantum circuit-based compression perspective for parameter-efficient learning. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=bB0OKNpznP>.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024d.
- Andrea Mari, Thomas R Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4:340, 2020.
- VP Masson-Delmotte, Panmao Zhai, SL Pirani, C Connors, S Péan, N Berger, Y Caud, L Chen, MI Goldfarb, and Pedro M Scheel Monteiro. Ipcc, 2021: Summary for policymakers. in: Climate change 2021: The physical science basis. contribution of working group i to the sixth assessment report of the intergovernmental panel on climate change. 2021.
- Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- Franco Molteni, Roberto Buizza, Tim N Palmer, and Thomas Petroliagis. The ecmwf ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society*, 122(529): 73–119, 1996.
- Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 2016.

- James O' Neill. An overview of neural network compression, 2020. URL <https://arxiv.org/abs/2006.03669>.
- Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight sharing. In *The mathematics of generalization*, pp. 373–394. CRC Press, 2018.
- Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.
- Jun Qi, Chao-Han Huck Yang, Pin-Yu Chen, and Min-Hsiu Hsieh. Theoretical error performance analysis for variational quantum circuit based functional regression. *npj Quantum Information*, 9(1):4, 2023.
- Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.
- Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.
- Dominik Seitz, Niklas Heim, João Moutinho, Roland Guichard, Vytautas Abramavicius, Aleksander Wenersteen, Gert-Jan Both, Anton Quelle, Caroline Groot, Gergana Velikova, Vincent Elfving, and Mario Dagrada. Qadence: a differentiable interface for digital and analog programs. *IEEE Software*, PP:1–14, 01 2025. doi: 10.1109/MS.2025.3536607.
- Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.
- Peter J Webster, Greg J Holland, Judith A Curry, and H-R Chang. Changes in tropical cyclone number, duration, and intensity in a warming environment. *Science*, 309(5742):1844–1846, 2005.
- Guangning Xu, Di Xian, Philippe Fournier-Viger, Xutao Li, Yunming Ye, and Xiuqing Hu. Am-convgru: a spatio-temporal model for typhoon path prediction. *Neural Computing and Applications*, 34(8):5905–5921, 2022.
- Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. Voice2series: Reprogramming acoustic models for time series classification. In *International conference on machine learning*, pp. 11808–11819. PMLR, 2021.
- Ming Ying, Wei Zhang, Hui Yu, Xiaoqin Lu, Jingxian Feng, Yongxiang Fan, Yongti Zhu, and Dequan Chen. An overview of the china meteorological administration tropical cyclone database. *Journal of Atmospheric and Oceanic Technology*, 31(2):287–301, 2014.

A QT APPROXIMATION ERROR

In the following analysis, we adopt the method introduced in (Qi et al., 2023) and extend its application to the QT framework. Assume the target NN with weight $W \in \mathbb{R}^m$ is formed with matrix $w \in \mathbb{R}^{l \times k}$ and bias $b \in \mathbb{R}^k$, such that $m = (k+1)l$, with sigmoid activation function. Given an input data $\mathbf{x} \in \mathbb{R}^l$, the prediction y for this model is

$$y(\mathbf{x}) = \sigma(w^T \mathbf{x} + b). \quad (41)$$

Define \mathbb{F}_{NN} as the classical NN hypothesis space, which represents a collection of NN operators. Given a set of N training data drawn independent and identically distributed from a data distribution \mathcal{D} , for a loss function ℓ and the smooth target function $h_{\mathcal{D}}^*$, defining the optimal NN operator $f_{\mathcal{D}}^* \in \mathbb{F}_{\text{NN}}$, the expected loss can be defined as:

$$\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\ell(h_{\mathcal{D}}^*(\mathbf{x}), f_{\mathcal{D}}^*(\mathbf{x}))], \quad (42)$$

and the empirical loss to minimize $\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)$ is defined as:

$$\mathcal{L}_{\mathcal{S}}(f_{\mathcal{D}}^*) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \ell(h_{\mathcal{D}}^*(\mathbf{x}_n), f_{\mathcal{D}}^*(\mathbf{x}_n)). \quad (43)$$

Now we further define $f_{\mathcal{D}}^*$ the optimal NN operator, f_S^* the empirical optimal operator, \bar{f}_S the returned NN operator. Without QT, using the error decomposition technique, the expected loss $\mathcal{L}_{\mathcal{D}}(\bar{f}_S)$ is factorized into:

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}(\bar{f}_S) &= \underbrace{\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)}_{\text{Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^*) - \mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)}_{\text{Estimation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f_S^*)}_{\text{Training Error}} \\ &\leq \mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*) + 2\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}}) + v \end{aligned} \quad (44)$$

where $\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}})$ is the empirical Rademacher complexity over the family \mathbb{F}_{NN} . To introduce the effect of QT, we define the \mathbb{F}_{QT} as the QT hypothesis space where $\mathbb{F}_{\text{QT}} \subset \mathbb{F}_{\text{NN}}$. Recall that \bar{f}_S is the returned NN hypothesis based on the training dataset S , in the case of QT, we define the \bar{f}_S^{QT} as the returned QT hypothesis while using QT, and $f_S^{\text{QT}*}$ the optimal QT operator. The QT approximation error is defined as $\mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*)$, this defines the error of how the optimal QT generated NN approximates the general target NN with arbitrary weight. While training process in QT involves only the update of θ_{qnn} and θ_{mm} , the training error is defined as $\mathcal{L}_{\mathcal{D}}(\bar{f}_S^{\text{QT}}) - \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*})$, indicating that the training is the process from \bar{f}_S^{QT} toward its optimal $f_S^{\text{QT}*}$.

In this sense, consider the expected loss $\mathcal{L}(\bar{f}_S^{\text{QT}})$ while using QT, the error decomposition technique reads:

$$\begin{aligned} \mathcal{L}(\bar{f}_S^{\text{QT}}) &= \underbrace{\mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)}_{\text{Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^*) - \mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*)}_{\text{Estimation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*)}_{\text{QT Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(\bar{f}_S^{\text{QT}}) - \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*})}_{\text{QT Training Error}} \\ &\leq \mathcal{L}_{\mathcal{D}}(f_{\mathcal{D}}^*) + 2\hat{\mathcal{R}}_S(\mathbb{F}_{\text{NN}}) + \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) + v \end{aligned} \quad (45)$$

while the bounds for approximation error and estimation error for classical NN should be known, and v refers to the training error that results from the optimization bias of gradient-based algorithms. We focus on the QT approximation error in the following.

We denote the weight and bias generated by QT in the classical NN model as w_{qt} and b_{qt} , such that

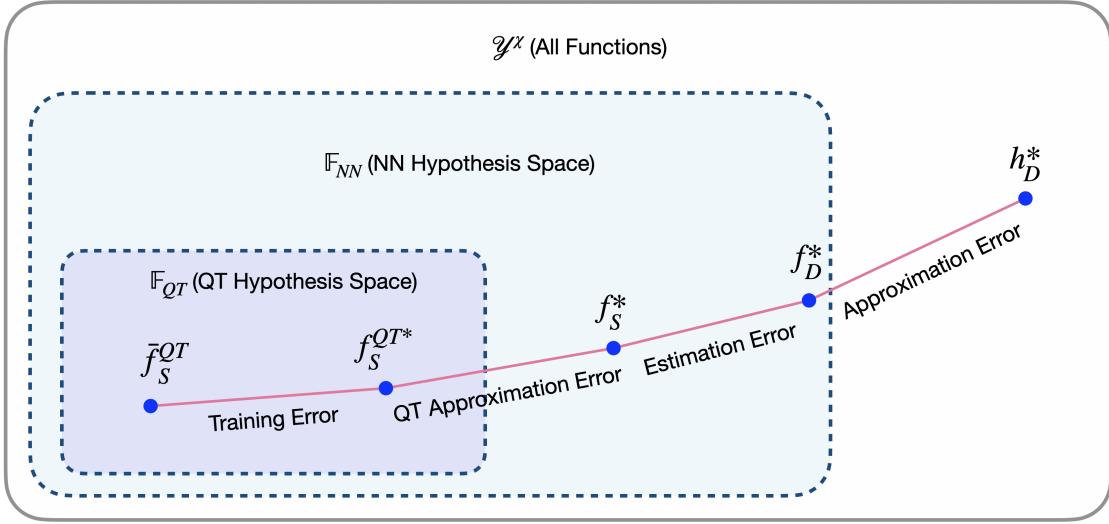
$$y_{\text{qt}}(\mathbf{x}) = \sigma(w_{\text{qt}}^T \mathbf{x} + b_{\text{qt}}), \quad (46)$$

where w_{qt} and b_{qt} are calculated from mapping model $G_{\theta_{\text{mm}}}$ with different parts of the basis set (equivalently different part of Ψ^c), denoted as

$$\begin{aligned} w_{\text{qt}} &= G_{\theta_{\text{mm}}}(\Psi_w^c(|\langle \phi | \psi(\theta_{\text{qnn}}) \rangle|^2)) \\ &= G_{\theta_{\text{mm}}}(\Psi_w^c(P(\theta_{\text{qnn}}))) \\ &= G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ P(\theta_{\text{qnn}}), \end{aligned} \quad (47)$$

where the measurement probability is represented as $P(\theta_{\text{qnn}})$ for clarity. Similarly,

$$\begin{aligned} b_{\text{qt}} &= G_{\theta_{\text{mm}}}(\Psi_b^c(|\langle \phi | \psi(\theta_{\text{qnn}}) \rangle|^2)) \\ &= G_{\theta_{\text{mm}}}(\Psi_b^c(P(\theta_{\text{qnn}}))) \\ &= G_{\theta_{\text{mm}}} \circ \Psi_b^c \circ P(\theta_{\text{qnn}}). \end{aligned} \quad (48)$$

Figure 15: **Error decomposition technique with QT.**

The QT approximation error can now be derived as:

$$\begin{aligned}
& \mathcal{L}_{\mathcal{D}}(f_S^{QT*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) \\
& \leq \|f_S^* - f_S^{QT*}\|_1 \\
& \leq \|f_S^* - \hat{f}_S^{QT*}\|_1 + \|\hat{f}_S^{QT*} - f_S^{QT*}\|_1 \quad (\text{Triangle Ineq.}) \\
& \approx \|f_S^* - \hat{f}_S^{QT*}\|_1 + \|\sigma'(c) \left((\hat{w}_{qt} - w_{qt})^T \mathbf{x} + \hat{b}_{qt} - b_{qt} \right)\|_1 \quad (\text{Mean Value Theorem}) \\
& \leq \|f_S^* - \hat{f}_S^{QT*}\|_1 + \frac{1}{4} \|(\hat{w}_{qt} - w_{qt})^T \mathbf{x} + \hat{b}_{qt} - b_{qt}\|_1 \\
& = \|f_S^* - \hat{f}_S^{QT*}\|_1 + \frac{1}{4} \left\| \left(G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ (\hat{P} - P) \right)^T \mathbf{x} + G_{\theta_{\text{mm}}} \circ \Psi_b^c \circ (\hat{P} - P) \right\|_1 \\
& \leq \|f_S^* - \hat{f}_S^{QT*}\|_1 + \frac{1}{4\sqrt{M}} \sqrt{\frac{1}{2} \ln \frac{2}{\delta}} \cdot \left\| (G_{\theta_{\text{mm}}} \circ \Psi_w^c(\mathbf{1}))^T \mathbf{x} + G_{\theta_{\text{mm}}} \circ \Psi_b^c(\mathbf{1}) \right\|_1 \quad (\text{Hoeffding's Ineq.}) \\
& = \|f_S^* - \hat{f}_S^{QT*}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \quad (\text{where } \|\mathbf{x}\| \leq \text{const.}) \\
& \leq \frac{1}{4} \|(w - \hat{w}_{qt})^T \mathbf{x} + b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& = \frac{1}{4} \|(w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ \hat{P})^T \mathbf{x} + b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& = \frac{1}{4} \|(w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | \psi(\theta_{\text{qnn}}) \rangle|_{\text{emp}}^2)^T \mathbf{x} + b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& = \frac{1}{4} \|(w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | U(\theta_{\text{qnn}}) | 0 \rangle^{\otimes n_{qt}}|_{\text{emp}}^2)^T \mathbf{x} + b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)
\end{aligned} \tag{49}$$

where \hat{f}_S^{QT*} is defined as the QT operator with finite measurement shots M , thus the probability \hat{P} is empirical, with the corresponding empirical \hat{w}_{qt} and \hat{b}_{qt} . c is a mean value vector of A and B where we use the property $\sigma(A) - \sigma(B) \approx \sigma'(c)(A - B)$ when $A - B$ is small, recall that $\sigma'(c) \leq \frac{1}{4}$. Continue on the QT

approximation error:

$$\begin{aligned}
& \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} + b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} \|_1 + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& = \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 + G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ (|\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 - |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2))^T \mathbf{x} \|_1 \\
& \quad + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} \|_1 \\
& \quad + \frac{1}{4} \| (G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ (|\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 - |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2))^T \mathbf{x} \|_1 \\
& \quad + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)
\end{aligned} \tag{50}$$

where the initial state $|\text{in}\rangle := |0\rangle^{\otimes n_{qt}}$, and S represented as an existing optimal operator that $U(\theta_{\text{qnn}})$ is targeting to approximate with the finite number of gate operations. Look into the term:

$$|\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 - |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2 \tag{51}$$

while considering S as $U(\theta_{\text{qnn}})$ with a perturbation, $S = U + V$ and $\|V\| \leq \epsilon_v$ with small ϵ_v , the term become:

$$\begin{aligned}
& |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 - |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2 \\
& = |\langle \phi | S - U | \text{in} \rangle|_{\text{emp}}^2 + 2\Re[\langle \phi | U | \text{in} \rangle \langle \text{in} | (S - U)^\dagger | \phi \rangle]_{\text{emp}}
\end{aligned} \tag{52}$$

With Solovay–Kitaev theorem $\|S - U\| \leq \epsilon$, the length L of the gate sequence needed to achieve this precision between the target unitary S and the approximated unitary U (constructed by gates from universal gate set \mathcal{G}), is given by $L = \mathcal{O}(\log^{c_{sk}}(1/\epsilon))$, we obtain

$$\begin{aligned}
& |\langle \phi | S - U | \text{in} \rangle|_{\text{emp}}^2 + 2\Re[\langle \phi | U | \text{in} \rangle \langle \text{in} | (S - U)^\dagger | \phi \rangle]_{\text{emp}} \\
& \leq \epsilon^2 + 2\epsilon \quad (\text{where } \|U\| = 1) \\
& \leq \epsilon(2 + \epsilon)
\end{aligned} \tag{53}$$

thus

$$\begin{aligned}
& \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} \|_1 \\
& \quad + \frac{1}{4} \| (G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ (|\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2 - |\langle \phi | U(\theta_{\text{qnn}}) | \text{in} \rangle|_{\text{emp}}^2))^T \mathbf{x} \|_1 \\
& \quad + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} \|_1 \\
& \quad + \frac{\epsilon(2 + \epsilon)}{4} \| (G_{\theta_{\text{mm}}} \circ \Psi_w^c(\mathbf{1}))^T \mathbf{x} \|_1 + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& \leq \frac{1}{4} \| (w - G_{\theta_{\text{mm}}} \circ \Psi_w^c \circ |\langle \phi | S | \text{in} \rangle|_{\text{emp}}^2)^T \mathbf{x} \|_1 + \frac{1}{4} \| b - \hat{b}_{qt} \|_1 \\
& \quad + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \quad (\text{Solovay–Kitaev Theorem})
\end{aligned} \tag{54}$$

where L is the gate sequence length, proportional to the depth of the QNN parameterized by θ_{qnn} , and constant $c_{sk} > 1$ is determined by the efficiency of the recursive refinement algorithm used to approximate a target

unitary S with a sequence of gates U from a universal set, where $c_{sk} \approx 3.97$ in the standard Solovay–Kitaev algorithm. In our context, a smaller c_{sk} means that fewer gates are required to approximate the given target unitary. Thus, it is natural to consider that c_{sk} is determined by the QNN architecture and can be reduced through QNN architecture search. Here, we use the behavior of this approximation to present the precision ϵ decreases exponentially with the sequence length $L^{1/c_{sk}}$.

Next, recall that the mapping model $G_{\theta_{\text{mm}}}$ is a classical NN with a sigmoid activation function. By the universal approximation theorem, such a network, with sufficient neurons, can approximate any continuous function on a compact domain, making it a universal approximator in principle (Barron, 1993). With $\prod_{k=1}^K h_k = h_g$ the width dependency of $G_{\theta_{\text{mm}}}$, the QT approximation error reads:

$$\begin{aligned}
& \mathcal{L}_{\mathcal{D}}(f_S^{\text{QT}*}) - \mathcal{L}_{\mathcal{D}}(f_S^*) \\
& \leq \frac{1}{4\sqrt{h_g}} \|\mathbf{1}\mathbf{x}\|_1 + \frac{1}{4} \|b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \quad (\text{Universal Approx.}) \\
& \leq \frac{1}{4} \|b - \hat{b}_{qt}\|_1 + \mathcal{O}\left(\frac{1}{\sqrt{h_g}}\right) + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \\
& \leq \mathcal{O}\left(\frac{1}{\sqrt{h_g}}\right) + \mathcal{O}\left(\frac{1}{\exp(L^{1/c_{sk}})}\right) + \mathcal{O}\left(\frac{1}{\sqrt{M}}\right)
\end{aligned} \tag{55}$$

where we apply the same procedure as \hat{w}_{qt} to \hat{b}_{qt} in the last step. Eq. 55 establishes an upper bound on the QT approximation error, incorporating key terms that contribute to the overall error performance of the QT framework. Specifically, it provides insights into how different factors—such as the width of the classical mapping model h_g , the number of quantum measurement shots M , and the depth of the QNN L —affect the error.