The Spectral Amplitude Principle for Dynamics of Quantum Neural Networks

Yihang Xu
School of Computer Science
Shanghai Jiao Tong University
Shenzhen Loop Area Institute
xyh_gb0608@sjtu.edu.cn

Dan-Bo Zhang*
School of Physics
South China Normal University
dbzhang@m.scnu.edu.cn

Junchi Yan*
School of Artificial Intelligence
School of Computer Science
Shanghai Jiao Tong University
Shanghai Innovation Institute
yanjunchi@sjtu.edu.cn

Abstract

The mechanism governing the training dynamics of Quantum Neural Networks (QNNs) remains under-explored. In classical Deep Neural Networks (DNNs), training is known dominated by "Spectral Bias," i.e. prioritizing learning low-frequency while struggling with high-frequency. QNNs exhibit similar spectral limitations when the target function possesses flat spectral amplitudes. However, in this work, considering target problems with different spectral amplitudes, we theoretically and empirically identify a distinct mechanism in QNNs, which we term *Spectral Amplitude Priority*. By analyzing the frequency-domain gradients and residual dynamics via the Quantum Neural Tangent Kernel (QNTK), we prove that QNN training is governed primarily by the magnitude of spectral components rather than their frequency indices. Consequently, QNNs can efficiently capture high-frequency functions—provided they have significant amplitude—thereby overcoming the inherent limitations of their classical counterparts. We validate this principle on both synthetic high-frequency functions and quantum-advantage tasks. These results show that QNNs notably outperform DNNs in high-frequency tasks, offering an explanation for QNNs' superior expressivity.

1 Introduction

Quantum machine learning (QML), often in the form of quantum neural networks (QNNs) [1] and embodied as parameterized quantum circuits (PQC) [2] has received intensive studies in recent years. In particular, efforts have been made to explore the quantum advantages for QML, in different aspects e.g. complexity [3, 4, 5], expressibility [6, 7], generalization [8] and trainability [9, 10].

One bottleneck of adopting QNNs is its notorious difficulty in training, where the so-called barren plateaus [11] problem remains open, although various mitigating methods have been devised: structured ansatz design [12, 13], initialization strategies [14, 15], and derivative-free optimization [16, 17, 18].

In this paper, we try to address another fundamental yet unresolved problem, i.e. understanding the training dynamics of QNNs. It not only relates to the aforementioned training stability problem,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Related works on spectral analysis in DNNs and QNNs.

Domain	Reference	Framework	Spectral Behavior	Primary Focus
DNNs	[24]	Empirical	Spectral Bias	Memorization & generalization phases
	[20]	Fourier Analysis		Robustness and convergence
	[25]	Linear Freq. Principle		Predicting learning dynamics
	[26]	Linear Freq. Principle		Linking error decay rate to activation
	[27]	Neural Tangent Kernel		Dynamics under non-uniform dist
	[28]	Neural Tangent Kernel		Mathematically linking between eigenvalues and convergence
	[29]	Partial Differential Equations		Role of initial dist. in frequency bias
QNNs	[22]	Trainable Frequency Modeling	Spectral Gap Modulation	Aligning with target spectrum
	[30]	Fourier Series Approximation	Truncated Spectrum	Evaluating ansatz capability for Fourier series
	[23]	Frequency Select Algorithms	Non-unique Frequency Map	Mitigating trainability issues via frequency selection
	Our Work	Analysis of gradient & residuals dynamics with QNTK	Spectral Amplitude Priority	Fitting priority is by spectral magnitude

but also sheds light on the expressibility of QNNs. For instance, whether or in what conditions, the QNN can capture the high-frequency spectral components of training data which is otherwise very difficult for its classical counterpart i.e. deep neural networks (DNNs) as already widely discussed in literature. Specifically, DNNs are known to be inherently limited by their low-frequency spectral preference [19, 20], a characteristic that notably extends to QNNs within the universal regime. In this work, we provide a systematic answer through both theoretical analysis and empirical evidence, showing that if target functions contain different spectral amplitudes, QNNs will enjoy better capability in capturing high-frequency components[2]. These results, to our best knowledge, have not been reported in literature, although there are some preliminary analyses [22, 23] from the frequency perspective.

Specifically, we show that QNNs follow a distinctive learning trajectory characterized by spectral amplitude priority. Our analysis is facilitated by establishing a gradient analysis framework to characterize the training mechanism. To further rigorously quantify the dynamics, we incorporate the quantum neural tangent kernels (QNTK) [31, 32] to derive an evolution equation for the training residual dynamics in the frequency domain. Crucially, we investigate scenarios with dominant high-frequency amplitudes to decouple the frequency index from signal strength, thereby distinguishing the amplitude priority mechanism from the frequency bias of DNNs.

Theoretically, we prove that (in Sec. 2.2.1) when the learning rate is reasonably small e.g. 0.01, the residuals corresponding to the high-amplitude spectral components decay quickly (i.e. exponentially) over training epochs[3]. In contrast, for classical DNNs, the model is confined to fit low-frequency information first, and it is challenging for them to capture the high-frequency part [19, 20]. These results indicate an advantage of QML as QNNs can enjoy superior expressiveness to fit with arbitrary frequency spectrum of the training data, as long as its corresponding amplitude is strong enough.
Our findings are:

• **Spectral Amplitude Priority Principle for QNNs:** We uncover the distinction between DNNs' known spectral bias (also called the frequency principle [19] i.e. favoring fitting the low-frequency part first in training) and QNNs' spectral amplitude principle namely the fitting is governed by the signal amplitude rather than frequency indices.

• **Expressiveness Advantages of QNNs:** High-frequency signals can be effectively learned when the corresponding spectral amplitude is strong, which indicates the potential advantage of QNNs. We further derive the closed-form expression for the gradients of the training loss in the frequency domain, providing explicit conditions for the frequency components prioritized during (early) training.

[2]For conciseness of analysis, the QNNs studied in this paper are grounded in the data re-uploading architecture [21] i.e. composed of an interleaved data encoding block and a trainable circuit block which is widely used in QNN literature.

[3]Empirically, although slight discrepancies between the analytical solution and actual evolution are observed in low-amplitude components during the initial training phase, the proposed residual dynamics framework yields a robust approximation of the overall training trajectory.

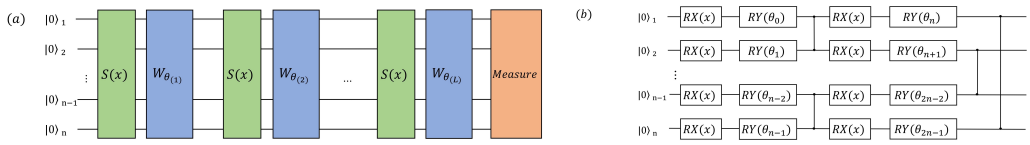


Figure 1: **The scheme of QNNs studied in the paper.** (a) $S(x)$ is the data encoding circuit block and $W_{\theta_{(p)}}$ is the trainable circuit block. (b) One of the layers of $S(x)$ and $W_{\theta_{(p)}}$ specific gates and their dependence on x, θ. The encoding circuit $S(x)$ embeds classical data into the rotation angles of RX gates, thereby encoding classical information into quantum state.

• **Residual Dynamics in Frequency Domain:** We reformulate the QNTK in frequency domain to derive a differential equation governing the training dynamics of residuals. This analytical framework provides a rigorous understanding that the residuals decay exponentially over training time.

• **Verification on Quantum Advantage Tasks:** Experiments on tasks with theoretically provable quantum advantages w.r.t. complexity and expressibility are performed. The results validate the existence of the spectral amplitude priority, as well as the performance superiority over their classical counterparts on simulation environments.

Remarks and Limitations. Existing studies [22, 23] for the training of QNNs are mostly confined to ideal quantum circuits as the noise can be complex which renders the theoretical analysis elusive. Likewise, we leave the experiments and further theoretical analysis for specific hardware environments for future work. As will be shown in detail in Sec. 2, our framework rests on two key assumptions. First, Our empirical validation relies on classical simulations of ideal state evolution. On real quantum processors, hardware constraints such as limited connectivity, gate infidelity, and compilation overhead may introduce deviations. Second, the Amplitude Priority mechanism is strictly derived from a gradient-based optimization perspective. Conversely, gradient-free optimization strategies, e.g., SPSA or Nelder-Mead may call for separate investigations. Table 1 categorizes existing literature and see Appendix A for details of related work.

2 Methods

We elucidate the Spectral Amplitude Principle in QNNs. First, we establish the gradient analysis framework to characterize the training mechanism in Sec. 2.1. Subsequently, we propose the QNTK in the frequency domain, and then derive the governing evolution equation for the residual dynamics in Sec. 2.2, providing a rigorous theoretical guarantee for the observed convergence behavior. Table 3 in Appendix provides a list of notations used to describe the QNN, the data re-uploading mechanism, frequency-domain analysis, and the residual analysis.

2.1 The Spectral Amplitude Principle for QNNs

2.1.1 Gradient Analysis in Frequency Domain

We focus on univariate function approximation to initially investigate the training dynamics of QNNs. Note that in this paper, our analysis is primarily conducted within the framework of the ideal circuit model, abstracting away hardware-specific overheads such as error correction and gate noise.

Without loss of generality, we follow the literature in QNNs training analysis, by employing the data-reuploading scheme [21] that alternatively sets the data encoding circuit block $S(x)$ and the trainable circuit block $W(\boldsymbol{\theta})$. As shown in Fig. 1, the data encoding block consists of gates of the form $S(x) = e^{-ixH}$, where H is an arbitrary Hamiltonian. $\boldsymbol{\theta} = (\theta_{(1)}, \theta_{(2)}, \ldots, \theta_{(L)})$ is the set of trainable parameters. The L-layer QNN is

$$U(x, \boldsymbol{\theta}) = W_{\theta_{(L)}} S(x) \cdots W_{\theta_{(2)}} S(x) W_{\theta_{(1)}} S(x). \tag{1}$$

The output state provides a way to express a function on x, where M is the measurement operator.

$$f(x, \boldsymbol{\theta}) = {}^{n\otimes}\langle 0|U^\dagger(x, \boldsymbol{\theta}) M U(x, \boldsymbol{\theta})|0\rangle^{\otimes n}. \tag{2}$$

The parameter θ is determined by minimizing the loss $L(\boldsymbol{\theta})$.

To formally analyze the training dynamics in frequency domain, the QNN output is expressed as a partial Fourier series [7, 33]:

$$\hat{f}(x, \boldsymbol{\theta}) = \sum_{\omega \in \Omega} C(\omega, \boldsymbol{\theta}) e^{i\omega \mathbf{x}}, \tag{3}$$

where the data encoding circuit determines the frequency ω and the rest of the circuit determines the coefficients $C(\omega, \boldsymbol{\theta})$ of the summation, and Ω denotes any accessible frequency as determined by the data encoding circuit module $S(x)$.

By Fourier transformation, we have

$$\hat{f}(k, \boldsymbol{\theta}) = \sum_{\omega \in \Omega} C(\omega, \boldsymbol{\theta}) \delta_{k,\omega}, \tag{4}$$

where the Dirac δ indicates that $\hat{f}(k, \boldsymbol{\theta})$ has a non-zero value only when k belongs to the accessible frequency.

As a generalization of [20, 34] in the classical setting, $\hat{y}(k)$ denotes the frequency components of the target function to learn, and the frequency-domain residual is defined as the difference between the QNN output spectrum and the target spectrum:

$$\hat{\varepsilon}(k, \boldsymbol{\theta}) = \hat{f}(k, \boldsymbol{\theta}) - \hat{y}(k), \tag{5}$$

We get the gradient of loss:

$$\frac{\partial \hat{L}(k)}{\partial \theta_j} = \hat{\varepsilon}(k, \boldsymbol{\theta}) \frac{\partial \hat{f}^{\dagger}(k, \boldsymbol{\theta})}{\partial \theta_j} + h.c. \tag{6}$$

where h.c. denotes the Hermitian conjugate of the preceding terms. The residual can be written as $\hat{\varepsilon}(k, \boldsymbol{\theta}) = A(k) e^{i\phi(k)}$:

$$\frac{\partial \hat{L}(k)}{\partial \theta_j} = A(k) \left[e^{i\phi(k)} \sum_{\omega \in \Omega} \frac{\partial C^{\dagger}(\omega, \boldsymbol{\theta})}{\partial \theta_j} \delta_{k,\omega} + h.c. \right]. \tag{7}$$

The analytical expression in Eq. 7 provides a fundamental insight into the training dynamics of QNNs on non-smooth data. Unlike classical DNNs, where the gradient magnitude typically decays with frequency (e.g., scaling as $\frac{1}{k^{\alpha}}$ due to the smoothness of activation functions), the derivative term $\frac{\partial \hat{f}}{\partial \theta}$ in QNNs remains *spectrally neutral*—it does not vanish as the frequency index k increases, provided $k \in \Omega$. This is because the coefficients $C(\omega, \boldsymbol{\theta})$ arise from norm-preserving unitary rotations that do not penalize high-frequency paths. Consequently, the gradient magnitude $\frac{\partial \hat{L}(k)}{\partial \theta_j}$ is linearly proportional to the residual amplitude $A(k)$, and the optimizer prioritizes frequency components with the largest residuals regardless of their frequency index. Crucially, at initialization, the QNN output is negligible due to random parameters, so $A(k) \approx |\hat{y}(k)|$—the initial gradient is directly proportional to the target's spectral amplitudes.

We term this phenomenon **Spectral Amplitude Priority**, which fundamentally differentiates QNN optimization dynamics from the Spectral Bias observed in classical networks.

2.1.2 Validation of Gradient Behavior

We follow the experimental setup as described in Sec. 3. To verify the Spectral Amplitude Principle, we design target functions with identical frequency bases $\{1, 3, 8\}$ but varying amplitude distributions. We select three target functions dominated by low, middle, and high frequencies, respectively:

$$\begin{aligned} f_L(x) &= 0.9\sin(x) + 0.1\sin(3x) + 0.1\sin(8x), \\ f_M(x) &= 0.1\sin(x) + 0.9\sin(3x) + 0.1\sin(8x), \\ f_H(x) &= 0.1\sin(x) + 0.1\sin(3x) + 0.9\sin(8x). \end{aligned} \tag{8}$$

We systematically investigate the convergence behavior of loss gradients for distinct frequency components. Fig. 2 visualizes gradient dynamics computed according to Eq. 7.

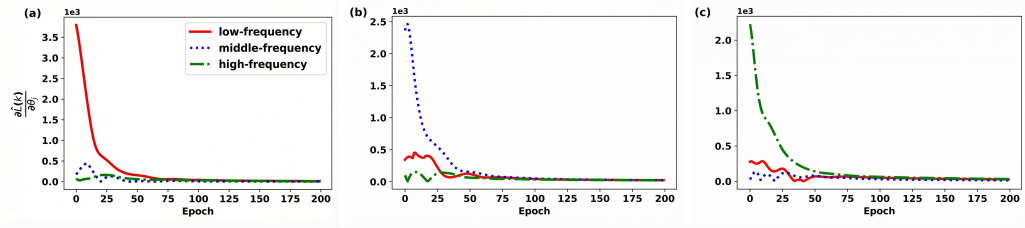


Figure 2: Evolutions of gradients derived from Eq. 7 for loss functions when training QNNs for fitting one-variable curves of low-frequency (a), middle-frequency (b) and high-frequency (c) dominated functions.

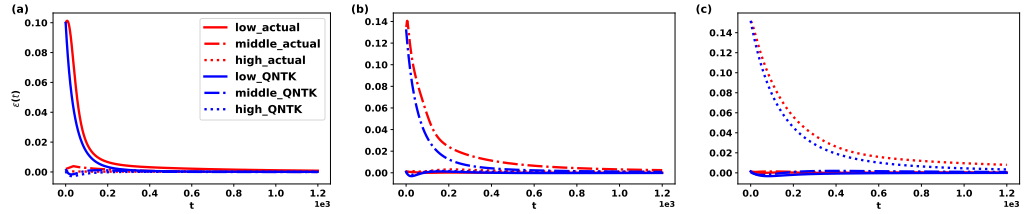


Figure 3: **Comparison of residual dynamics between actual ones and predicted by QNTK when learning low-frequency (a), middle-frequency (b) and high-frequency (c) dominated functions.** We plot the actual dynamics of $\varepsilon(t)$ for the above three functions, taking three peak frequencies in frequency domain corresponding to low, medium, and high frequencies, and their $\varepsilon(t)$ theoretical predictions of QNTK. The qubit number is 8 with learning rate $\eta = 0.01$. Parameters in the ansatz are initialized randomly in $[0, 2\pi]$.

The results show that QNNs adaptively prioritize the convergence of frequency components with high amplitudes. Specifically, when the target function is high-frequency dominated ($f_H(x)$), the gradients for high-frequency components—computed via Eq. 7—exhibit significant magnitudes immediately upon initialization, thereby circumventing the spectral suppression typically observed in DNNs. This confirms that the training dynamics of QNNs are fundamentally governed by amplitude priority, providing robust empirical support for the Spectral Amplitude Principle.

2.2 Residual Dynamics in Frequency Domain

We now investigate the training dynamics by capturing the average behavior of the residual dynamics at different frequencies in the frequency domain.

2.2.1 Residual Analysis Via QNTK

In line with the setting [31, 32] in x-space, the difference equation for the residual dynamics (Eq. 5) derived from the loss gradient is expressed as follows, where the variable $\boldsymbol{\theta}$ is omitted for brevity:

$$\Delta\varepsilon(x_{i'}) \approx \sum_j \frac{\partial \varepsilon(x_{i'})}{\partial \theta_j} d\theta_j = -\sum_i \eta \mathcal{K}_{\boldsymbol{\theta}}(x_{i'}, x_i)\varepsilon(x_i),$$

where $\mathcal{K}_{\boldsymbol{\theta}}(x_{i'} x_i) = \sum_j \frac{\partial d\varepsilon(x_{i'})}{\partial \theta_j} \frac{\partial d\varepsilon(x_i)}{\partial \theta_j}$. The derivation of the residual dynamics operates strictly within the lazy training regime [31]. This theoretical framework relies on the assumption of a sufficiently small learning rate, which effectively confines the parameters to a small neighborhood of their initialization. Such an approximation allows the QNTK to be treated as a time-invariant constant during training, i.e., $\mathcal{K}(x_{i'}, x_i) \approx \overline{\mathcal{K}}(x_{i'}, x_i)$. We empirically verify the validity of this lazy-training assumption by quantitatively measuring the QNTK drift throughout training; the detailed measurement protocol and results are deferred to Appendix B. Here

$$\overline{\mathcal{K}}(x_{i'}, x_i) = \frac{2L(D|\chi_{x_{i'}, x_i}|^2 - 1)}{(D^2 - 1)^2}(D\mathrm{Tr}(M^2) - (\mathrm{Tr}M)^2),$$

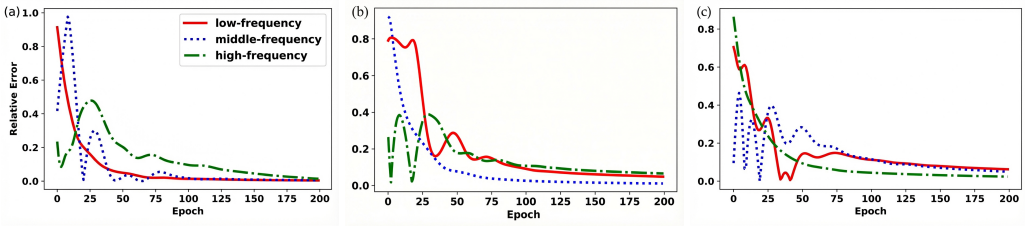


Figure 4: Evolutions of relative errors for low-frequency (a), middle- (b), and high- (c) dominated functions over iterations.

We can apply the Fourier transformation and write the difference equation for residual dynamics in the frequency domain as,

$$\Delta\varepsilon(k) = \sum_i -\eta\overline{\mathcal{K}}(k,k')\varepsilon(k'), \tag{9}$$

where the Fourier transform of QNTK is,

$$\overline{\mathcal{K}}(k',k) = \frac{1}{N}\sum_{ii'}\overline{\mathcal{K}}(x_{i'},x_i)e^{ik'x_{i'}}e^{-ikx_i}.$$

For the above linear difference Eq. (9), we can directly solve the linear equation to obtain the solution (for simplicity frequency k has been omitted here):

$$\varepsilon(t) = e^{-\eta\overline{\mathcal{K}}t}\varepsilon(0), \tag{10}$$

where $\varepsilon(0)$ denotes the residual at t = 0.

Eq. 10 gives exponential decay along the eigenmodes of $\overline{\mathcal{K}}$, with rates set by its eigenvalues. To examine individual Fourier components, let $\mathbf{e}_k$ select frequency k and define its Rayleigh strength as $s_k = \mathbf{e}_k^\dagger\overline{\mathcal{K}}\mathbf{e}_k = \overline{\mathcal{K}}(k,k)$. This quantity measures the self-coupling of the kth Fourier component in the frequency-domain QNTK. Over 40 seeds and 321 grid points, the measured strengths are $s_1 = 0.407$, $s_3 = 0.385$, and $s_8 = 0.274$. Under the diagonal control, $\varepsilon(k,t) = (1-\eta s_k)^t\varepsilon(k,0)$, so each component has the normalized decay factor $|1-\eta s_k|^t$. The ordering $s_1 > s_3 > s_8$ therefore predicts $k = 1$ as the fastest-decaying component for every target-amplitude permutation. The full Fourier-component update also contains cross-frequency coupling:

$$\varepsilon(k,t+1) = \varepsilon(k,t) - \eta\left[s_k\varepsilon(k,t) + \sum_{k'\neq k}\overline{\mathcal{K}}(k,k')\varepsilon(k',t)\right]. \tag{11}$$

The cross-terms depend on the residual amplitudes at the other frequencies and couple their Fourier-component trajectories.

2.2.2 Validation of Residual Behavior

We contrast the analytical predictions derived from the QNTK regime with the actual training dynamics observed during the optimization of the three target functions defined in Eq. 8 to empirically validate our framework.

As illustrated in Fig. 3, the analytical solutions exhibit a high degree of fidelity to the empirical residual dynamics. The residuals of frequency components with significant spectral amplitudes decay exponentially under a small learning rate η. The analytic solutions do not fit well with those of actual dynamics for the remaining low-amplitude components at the early stage; nevertheless, the framework still provides a robust approximation of the global training process since those residuals are several orders of magnitude smaller.

3 Experiments

To validate the proposed Spectral Amplitude Principle without the interference of hardware-specific noise and connectivity constraints which could potentially mask the intrinsic training dynamics we conduct our experiments in an ideal, noise-free environment.

All the simulations are performed on a classical machine with an Intel Core i7-14700HX CPU and an NV GeForce RTX 4060 GPU. We implement a Python quantum simulator based on PennyLane [35].

Table 2: Comparison of DNNs with different dimensions in fitting high-frequency functions.

Config (Hidden Layers)	Params	Train Acc.	Test Acc.
[200, 200, 200, 100]	100,300	0.99	0.75
[50, 50]	2,600	0.63	0.62
[100, 100]	10,200	0.61	0.60
[300, 300, 300]	180,600	0.64	0.56
[500, 500, 500, 500]	751,000	0.87	0.70

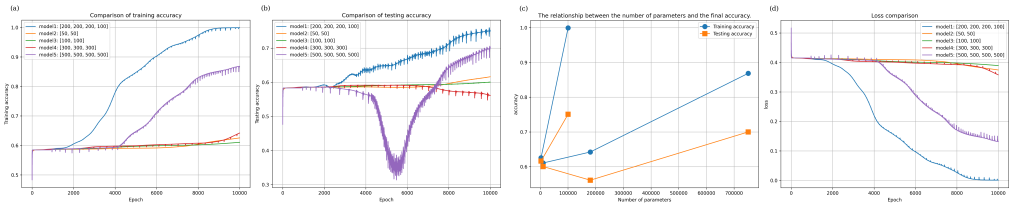


Figure 5: **Performance analysis of various DNN architectures on the high-frequency task.** (a-b) Evolution of training and test accuracy over 10K epochs for different configurations. (c) The relationship between model parameters and accuracy. (d) Comparison of training loss trajectories across the evaluated models.

3.1 Preliminary Experiments

We first examine the optimization trajectories of three univariate target functions to fit, as listed in Eq. 8 for a preliminary study. The quantum circuit comprises 4 qubits and a depth of 20 layers, with a total of 160 trainable parameters. The data encoding module uses angle encoding.

3.1.1 Example of Spectral Amplitude Priority

Fig. 4 tracks the relative error evolution for all frequency components in QNNs. The results establish that the rate of error convergence is primarily dictated by the magnitude of spectral coefficients—components with significant amplitudes consistently converge fastest, invariant to the frequency regime. In the high-frequency dominated case (Fig. 4 (c)), the high-frequency mode is prioritized precisely because it carries the largest spectral weight.

For the amplitude permutations in Eq. 8, the observed QNN prioritizes $k = 1, 3, 8$ for f_L, f_M, f_H, respectively, and the full QNTK reproduces this sequence. The diagonal control prioritizes $k = 1$ for all three targets, consistent with $s_1 > s_3 > s_8$. Thus, the cross-frequency terms in Eq. 11 produce the amplitude-conditioned reversal of the learning order. We quantify their contribution using a frozen empirical QNTK: the full-matrix prediction retains every $\overline{\mathcal{K}}(k, k')$, and the diagonal control sets the cross-frequency entries to zero while keeping the diagonal entries, initial residuals, and learning rate fixed. Comparing the predicted relative-error ordering with the observed QNN ordering over epochs 0–20 gives ranking accuracies of 94.5% for the full matrix and 29.7% for the diagonal control.

To verify that Spectral Amplitude Priority is not an artifact of a specific QNN configuration, we further conduct ablation studies on $f_H(x)$ along three architectural axes: circuit depth $L \in \{8, 12, 16, 20\}$, qubit count $n \in \{2, 4, 8\}$, and data encoding ($R_Y(x)$, $R_Y(x/2)$, $R_Y(x)+R_X(x)$). Across all 10 configurations, the highest-amplitude component ($k=8$) consistently converges first with smooth monotonic decay and small variance, whereas the low-amplitude components ($k = 1, 3$) remain oscillatory and seed-sensitive. Full results are provided in Appendix C.

3.1.2 Comparison with Classical DNNs

We benchmark QNNs against DNNs to validate their distinct inductive biases. To ensure a rigorous comparison, we perform an architectural sweep to identify the most competitive classical baseline. As summarized in Table 2, we sweep DNN architectures with parameter counts spanning 2.6×10^3 to 7.5×10^5 (tanh activation). The configuration [200, 200, 200, 100] yields the best test accuracy (0.75), yet further increasing capacity (e.g., [500, 500, 500, 500]) degrades generalization to 0.70, confirming that high-frequency learning difficulty cannot be resolved by over-parameterization alone.

The training and test dynamics for these varying DNN scales are further elucidated in Fig. 5. The accuracy is measured by $1 - L(\boldsymbol{\theta})$. Increasing the parameter budget fails to yield a proportional

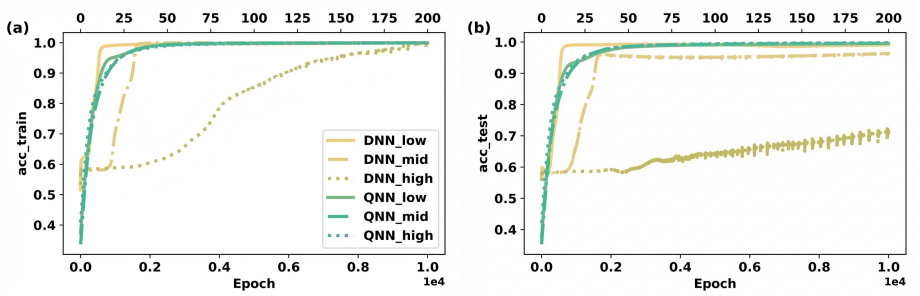


Figure 6: **Comparison of accuracy of fitting (a) and predicting (b) between DNNs and QNNs for low-frequency, middle-frequency and high-frequency dominated functions, respectively.** The bottom x-axis indicates DNN training epochs (10,000) and the top indicates QNN training epochs (200).

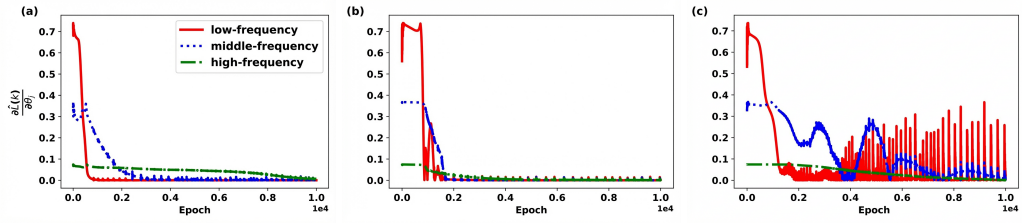


Figure 7: Evolutions of gradients for loss of DNNs in fitting low-frequency (a), middle- (b) and high-functions (c).

improvement in test accuracy for high-frequency function. Crucially, our comparative analysis in Table 2 and Fig. 5 focuses specifically on the high-frequency dominated function ($f_H(x)$). This choice is motivated by the fact that DNNs already exhibit efficient convergence in the low- and middle-frequency components.

We select this best DNN configuration (approx. 10^5 parameters) as the classical baseline. As shown in Fig. 6, while the DNN achieves high training accuracy across all target functions via memorization, it fails to generalize on $f_H(x)$. In contrast, the QNN with two orders of magnitude fewer parameters shows robust generalization across the entire frequency spectrum. In Fig. 7 The gradient dynamics confirms that DNN optimization adheres strictly to spectral bias—even for $f_H(x)$, high-frequency gradients remain suppressed—in sharp contrast to the amplitude-governed QNN dynamics shown in Fig. 2. We further verify that this limitation persists across activation functions (ReLU, Sigmoid, hat); results are in Appendix D.

3.1.3 Comparison with Spectral-Bias-Mitigating Architectures

We further benchmark QNNs against three classical architectures designed to alleviate spectral bias: Fourier Feature Networks (FFN) [36] with scale parameter σ, SIREN [37] with frequency-scaling ω_0, and KAN [38] with learnable spline grid size. All baselines are configured with $\approx$160 parameters to match the QNN budget.

As shown in Fig. 8, all three architectures exhibit pronounced hyperparameter dependence with a common qualitative pattern: the model fails to fit the high-frequency target unless its architectural prior is tuned into the appropriate spectral or resolution regime. FFN displays a "U-shaped" curve where performance degrades at both extremes of σ; SIREN shows a sharp phase transition requiring $\omega_0 \geq 3$ to escape the low-frequency regime; KAN needs sufficiently fine grids to resolve high-frequency components.

These architectures do not overcome spectral bias through intrinsic training dynamics but rather inject high-frequency capacity via target-dependent hyperparameters. In contrast, the QNN attains a consistently low test MSE (red dashed line) without any frequency-related hyperparameter, confirming that Spectral Amplitude Priority is an intrinsic dynamical property rather than an architectural prior. Extended discussion is provided in Appendix E.

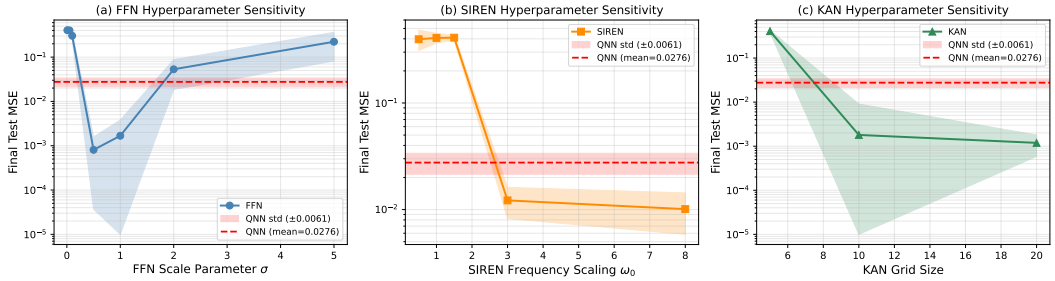


Figure 8: **Hyperparameter sensitivity of three spectral-bias-mitigating architectures (≈ 160 parameters).** (a) FFN: test MSE vs. Fourier scale σ. (b) SIREN: test MSE vs. ω_0. (c) KAN: test MSE vs. grid size. Shaded regions: one std across seeds. Red dashed line: QNN mean performance (≈ 0.0276) without any frequency-scaling hyperparameter.

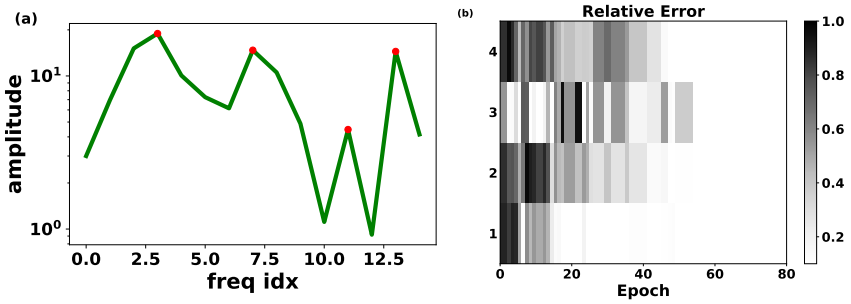


Figure 9: **Result of learning the Iris dataset with QNNs in the frequency domain.** (a) Magnitude-frequency map. (b) Evolution of relative errors at different frequencies during training. Vertical coordinates '1', '2', '3', '4' indicate the relative error of the selected frequency samples. The indices corresponding to the frequencies increase sequentially from the bottom to the top.

3.2 Experiments on Classification

We consider the classification of Iris dataset $\{(x_i, y_i)\}_{i=0}^{N-1}$. Regarding the theoretical foundation of this advantage, it has been mathematically established that classically simulating the output probability distributions of Instantaneous Quantum Polynomial circuits—typically characterized by a Hadamard-Diagonal-Hadamard structure—requires exponential overhead. Conversely, quantum processors can efficiently implement these circuits by simply executing the gate layers, thereby showing a provable quantum speedup.

We adopt an ansatz with 2 qubits and 6 layers. The Mean Squared Error (MSE) is used as the loss function.

Fig. 9 (a) presents the magnitude-frequency map computed via the Discrete Fourier Transform, revealing significant non-uniformity in spectral amplitudes with a distinct peak structure. Red markers identify representative frequency components spanning both dominant and secondary amplitudes. Fig. 9 (b) shows the temporal evolution of relative errors for these selected components, where color intensity encodes error magnitude (black: high error; white: low error). The results confirm that higher-amplitude components converge preferentially.

3.3 Experiments on Discrete Logarithm Problem

We further validate the Spectral Amplitude Priority on the Discrete Logarithm Problem (DLP) [39], whose modular exponentiation exhibits extreme non-linearity with energy predominantly dispersed across high-frequency components (Fig. 10 (a)). This high-frequency-dominant spectral structure makes DLP an ideal testbed. Notably, the quantum advantage for DLP has been rigorously proved [5]: classical algorithms provably cannot learn the classification rule, whereas quantum kernel methods achieve high accuracy in polynomial time. While the original work uses a given quantum kernel, here we adopt a variational quantum circuit to parameterize the quantum kernel. This allows us to incorporate DLP tasks into the training framework of QNNs.

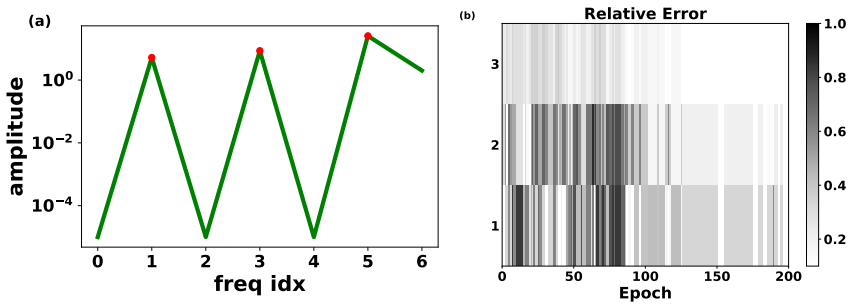


Figure 10: **Result of learning the discrete logarithm problem with QNNs in the frequency domain.** (a) Magnitude-frequency map. (b) Evolution of relative errors at different frequencies during training. The meanings of '1', '2' and '3' in the vertical coordinates are the same as in Fig. 9 (b).

We consider the DLP over the finite multiplicative group Z_p^*, defined by a large prime p and a generator α. The task is to recover the integer x (where $0 \leq x \leq p-2$) satisfying $\alpha^x \equiv \beta (\text{mod } p)$ denoted as $x = \log_\alpha \beta$. We conduct a labeled dataset based on discrete logarithmic properties.

Our setup uses 8 qubits and 24 layers with data re-uploading, applied to 40 samples. We employ a variational quantum kernel $K_{ij} = |\langle \psi(x_i) | \psi(x_j) \rangle|^2$ optimized via kernel-target alignment [40] prior to SVM training.

As shown in Fig. 10 (b), the relative error decreases preferentially for frequencies with the highest spectral peaks. Lower-amplitude components are subsequently learned in descending order of their spectral contributions, empirically confirming that Spectral Amplitude Priority governs QNN dynamics on quantum-advantage tasks.

4 Conclusion

We have established Spectral Amplitude Priority as the fundamental mechanism of QNN training dynamics. We show that considering target problems with different spectral amplitudes, the spectral fitting priority of QNNs is dictated by the magnitude of spectral coefficients rather than their frequency indices. This mechanism enables QNNs to excel in high-frequency dominant regimes where classical DNNs typically fail due to spectral bias.

5 Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

6 Acknowledgement

This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM411), Shanghai Municipal Commission of Economy and Informatization (2025-GZL-RGZN-BTBX-01006), and the National Natural Science Foundation of China (Grants No. 12375013 and No. 12547109).

References

[1] Vedran Dunjko and Hans J Briegel. Machine learning and artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 81(7):074001, jun 2018.

[2] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. Quantum Science and Technology, 4(4):043001, nov 2019.

[3] 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.

[4] Dan-Bo Zhang, Shi-Liang Zhu, and Z. D. Wang. Protocol for implementing quantum nonparametric learning with trapped ions. Phys. Rev. Lett., 124:010506, Jan 2020.

[5] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. Nature Physics, 17:1013–1017, Sep 2020.

[6] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. Efficient measure for the expressivity of variational quantum algorithms. Phys. Rev. Lett., 128:080506, Feb 2022.

[7] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. Phys. Rev. A, 103:032430, Mar 2021.

[8] 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.

[9] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, Shan You, and Dacheng Tao. Learnability of quantum neural networks. PRX quantum, 2(4):040337, 2021.

[10] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature communications, 12(1):1791, 2021.

[11] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. Nature communications, 9(1):4812, 2018.

[12] Kaining Zhang, Min-Hsiu Hsieh, Liu Liu, and Dacheng Tao. Toward trainability of quantum neural networks. arXiv preprint arXiv:2011.06258, 2020.

[13] AV Uvarov and Jacob D Biamonte. On barren plateaus and cost function locality in variational quantum algorithms. Journal of Physics A: Mathematical and Theoretical, 54(24):245301, 2021.

[14] Kaining Zhang, Liu Liu, Min-Hsiu Hsieh, and Dacheng Tao. Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits. Advances in Neural Information Processing Systems, 35:18612–18627, 2022.

[15] Chae-Yeun Park and Nathan Killoran. Hamiltonian variational ansatz without barren plateaus. Quantum, 8:1239, 2024.

[16] Yichen Xie. Quantum surrogate-driven image classifier: A gradient-free approach to avoid barren plateaus. arXiv preprint arXiv:2505.05249, 2025.

[17] Jakab Nádori, Gregory Morse, Barna Fülöp Villám, Zita Majnay-Takács, Zoltán Zimborás, and Péter Rakyta. Batched line search strategy for navigating through barren plateaus in quantum circuit training. Quantum, 9:1841, 2025.

[18] Ernesto Acosta, Carlos Cano Gutiérrez, Guillermo Botella, and Roberto Campos. Adiabatic training for variational quantum algorithms. In European Conference on Parallel Processing, pages 88–99, 2024.

[19] Zhi-Qin John Xu. Frequency principle in deep learning with general loss functions and its potential application. arXiv preprint arXiv:1811.10146, 2018.

[20] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In International conference on machine learning, pages 5301–5310, 2019.

[21] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. Quantum, 4:226, February 2020.

[22] Ben Jaderberg, Antonio A Gentile, Youssef Achari Berrada, Elvira Shishenina, and Vincent E Elfving. Let quantum neural networks choose their own frequencies. Physical Review A, 109(4):042421, 2024.

[23] Michael Poppel, David Bucher, Maximilian Zorn, Nico Kraus, Philipp Altmann, Jonas Stein, and Claudia Linnhoff-Popien. Mitigating exponential mixed frequency growth through frequency selection. arXiv preprint arXiv:2508.10533, 2025.

[24] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Proceedings of the 34th International Conference on Machine Learning, volume 70, pages 233–242, 06–11 Aug 2017.

[25] Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. Explicitizing an implicit bias of the frequency principle in two-layer neural networks. arXiv preprint arXiv:1905.10264, 2019.

[26] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. On the exact computation of linear frequency principle dynamics and its generalization. SIAM Journal on Mathematics of Data Science, 4(4):1272–1292, 2022.

[27] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In International conference on machine learning, pages 685–694, 2020.

[28] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pages 2205–2211, 8 2021.

[29] Juan Molina, Mircea Petrache, Francisco Sahli Costabal, and Matías Courdurier. Understanding the dynamics of the frequency bias in neural networks. arXiv preprint arXiv:2405.14957, 2024.

[30] Dirk Heimann, Hans Hohenfeld, Gunnar Schönhoff, Elie Mounzer, and Frank Kirchner. Learning fourier series with parametrized quantum circuits. Physical Review Research, 7(2):023151, 2025.

[31] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. Analytic theory for the dynamics of wide quantum neural networks. Phys. Rev. Lett., 130:150601, Apr 2023.

[32] Junyu Liu, Francesco Tacchino, Jennifer R Glick, Liang Jiang, and Antonio Mezzacapo. Representation learning via quantum neural tangent kernels. PRX Quantum, 3:030323, Aug 2022.

[33] Berta Casas and Alba Cervera-Lierta. Multidimensional fourier series with quantum circuits. Physical Review A, 107(6), Jun 2023.

[34] Zhi-Qin John Xu. Frequency principle: Fourier analysis sheds light on deep neural networks. Communications in Computational Physics, 28(5):1746–1767, June 2020.

[35] Ville Bergholm, Josh A. Izaac, Maria Schuld, Christian Gogolin, and Nathan Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968, 2018.

[36] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in neural information processing systems, 33:7537–7547, 2020.

[37] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems, 33:7462–7473, 2020.

[38] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. arXiv preprint arXiv:2404.19756, 2024.

[39] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. On the discrete logarithm problem in finite fields of fixed characteristic. Transactions of the American Mathematical Society, 370(5):3129–3145, 2018.

[40] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan HS Derks, Paul K Faehrmann, and Johannes Jakob Meyer. Training quantum embedding kernels on near-term quantum computers. Physical Review A, 106(4):042431, 2022.

[41] Francesco Scala et al. Towards practical quantum neural network diagnostics with neural tangent kernels. arXiv preprint arXiv:2503.01966, 2025.

Table 3: List of symbols and notations.

Symbol	Description
x	Input data
$\boldsymbol{\theta}$	Trainable parameters of QNN
$S(x)$	Data encoding block
$W(\boldsymbol{\theta})$	Trainable circuit block
$f(x, \boldsymbol{\theta})$	QNN output function
$\hat{f}(k, \boldsymbol{\theta})$	Fourier coefficient at frequency k
Ω	Set of accessible frequencies
$\hat{\varepsilon}(k)$	Residual in frequency domain
$A(k)$	Amplitude of the residual
η	Learning rate
$\mathcal{K}(x, x')$	QNTK
$\overline{\mathcal{K}}(k, k')$	Frequency-domain QNTK

A Related Work

The training dynamics of DNNs have evolved from empirical observations to rigorous theoretical formalizations. Early research has identified a pervasive "Spectral Bias," where the classical neural networks prioritize low-frequency components during the early training phases [24, 20]. To theoretically ground these observations, subsequent studies develop dynamic models such as the Linear Frequency-Principle [25, 26]. This theoretical study culminates in the application of the Neural Tangent Kernel (NTK) framework, where convergence speeds are mathematically linked to the eigendecomposition of the kernel [27, 28] and further refined by analyzing the initialization distributions via partial differential equations [29]. Consequently, the spectral dynamics of classical models are now well-charted.

In the quantum domain, Fourier analysis has also emerged as a fundamental tool, primarily for characterizing the expressivity of PQC (namely QNN) [7]. Recent methodologies have focused on manipulating the frequency spectrum to enhance model capacity. Notable approaches include introducing trainable-frequency models to capture non-regularly spaced frequencies [22], defining metrics to evaluate ansatz capabilities in approximating Fourier series [30], and developing selection algorithms to mitigate the curse of dimensionality in multi-dimensional inputs [23].

However, these works primarily address the question of which frequencies are representable by the QNN, leaving the question of in what order frequencies are learned largely unexplored. Our work fills this gap by characterizing the spectral fitting (or convergence) dynamics during training.

B Empirical Verification of the Lazy Training Regime

The residual dynamics derived in Sec. 2.2.1 rely on the lazy training regime assumption, which requires the QNTK to remain approximately constant throughout the optimization process. To rigorously substantiate this assumption underlying our theoretical framework, we explicitly quantify the temporal drift of the QNTK during training.

Measurement Methodology. We characterize the QNTK drift by tracking the relative change of its Frobenius norm with respect to the initialization, defined as

$$\Delta_{\mathcal{K}}(t) \;=\; \frac{\|\mathcal{K}(t) - \mathcal{K}(0)\|_F}{\|\mathcal{K}(0)\|_F}, \tag{12}$$

where $\mathcal{K}(t)$ denotes the empirical QNTK matrix evaluated at training step t. The measurement is conducted under the same setting as Sec. 3, with a learning rate of $\eta = 0.01$ and a total of $1{,}200$ training epochs.

Empirical Results. Table 4 reports the relative drift $\Delta_{\mathcal{K}}(t)$ at representative training epochs, while Fig. 11 visualizes the full evolution of the drift trajectory throughout the optimization process. Across the entire training horizon, the maximum observed drift is 6.22% and the mean drift is 3.32%. As clearly illustrated in Fig. 11, the drift exhibits a rapid initial increase during the early training phase, followed by a pronounced saturation plateau after roughly 300 epochs, beyond which $\Delta_{\mathcal{K}}(t)$ remains nearly stationary. Both the maximum and mean drift remain well below the 10% threshold, which

is a conventional empirical boundary established in the literature to certify the validity of linear approximations within the lazy training regime [41].

Table 4: Relative QNTK drift $\Delta_{\mathcal{K}}(t)$.

Epoch	$\Delta_{\mathcal{K}}(t)$	**Epoch**	$\Delta_{\mathcal{K}}(t)$
0	0.0005	100	0.0360
5	0.0031	150	0.0452
10	0.0056	300	0.0576
20	0.0106	500	0.0579
30	0.0151	800	0.0583
50	0.0226	1000	0.0600
75	0.0300	1199	0.0622

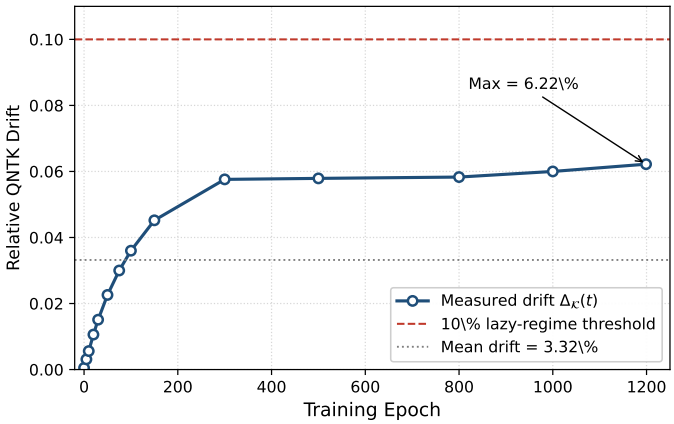


Figure 11: Relative Frobenius drift of the QNTK during training. The measured drift $\Delta_{\mathcal{K}}(t)$ (blue solid line with markers) remains consistently below the conventional 10% lazy-regime threshold (red dashed line) over $1{,}200$ training epochs, with a mean drift of 3.32% (gray dotted line) and a maximum of 6.22%. The pronounced saturation plateau observed after approximately 300 epochs empirically validates the time-invariant approximation of the QNTK adopted in our theoretical analysis.

Discussion. These measurements indicate that the QNTK is effectively confined to a small neighborhood of its initialization throughout the training process. Consequently, the time-invariant approximation $\mathcal{K}(x_{i'}, x_i) \approx \overline{\mathcal{K}}(x_{i'}, x_i)$ adopted in Eq. 9 is empirically justified. This stability of the QNTK provides a reliable foundation for the closed-form residual dynamics in Eq. 10, and thereby supports the rigor of the Spectral Amplitude Priority analysis derived therefrom.

C Ablation Studies on Architectural Hyperparameters

The theoretical analysis in Sec. 2 establishes that QNN convergence is governed by the spectral amplitude rather than the frequency index of the target. To examine whether this conclusion is robust to the specific architectural choices used in our preliminary experiments, this appendix reports ablation studies along three orthogonal dimensions: circuit depth L, qubit count n, and data encoding strategy. All experiments adopt the high-frequency-dominated target

$$f_H(x) = 0.1\sin(x) + 0.1\sin(3x) + 0.9\sin(8x),$$

which is deliberately constructed so that the highest-frequency component ($k = 8$) carries the dominant spectral amplitude, while the low-frequency components ($k = 1, 3$) carry substantially smaller amplitudes. This setting maximally decouples the frequency index from the signal strength and thus provides a stringent test of Spectral Amplitude Priority. For every configuration, results are averaged over multiple independent random seeds, and we report the per-frequency relative error $|\hat{\varepsilon}(k,t)|/|\hat{y}(k)|$ as a function of training epoch.

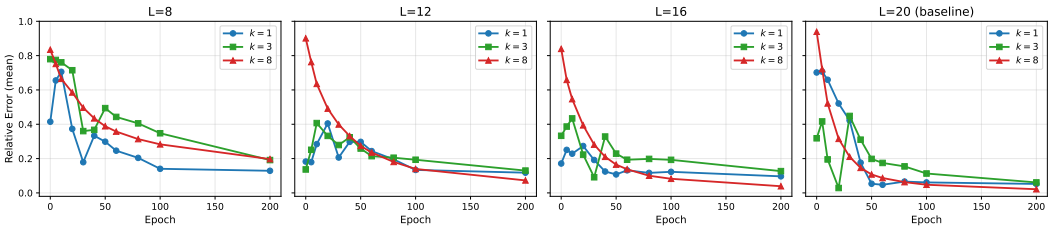


Figure 12: **Ablation on circuit depth.** Mean relative error of the three frequency components ($k = 1, 3, 8$) over training epochs for $L \in \{8, 12, 16, 20\}$. The highest-amplitude component $k = 8$ consistently converges first across all depths.

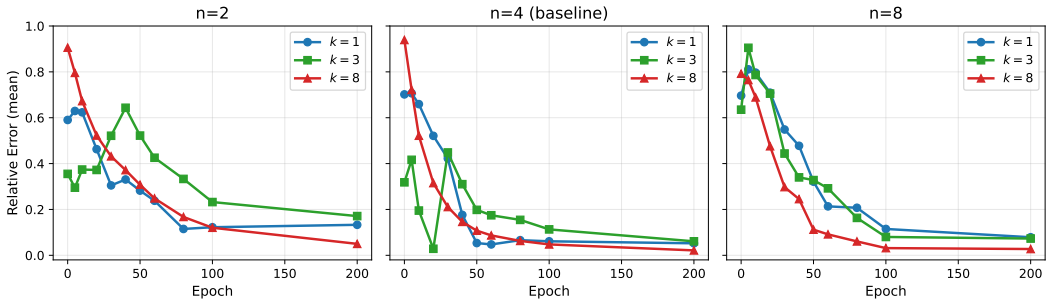


Figure 13: **Ablation on qubit count.** Mean relative error over training epochs for $n \in \{2, 4, 8\}$. The convergence ordering $k = 8 \succ k = 3 \succ k = 1$ persists irrespective of qubit count.

Two consistent phenomena emerge across all three ablation dimensions and serve as the empirical signature of Spectral Amplitude Priority:

- **Early-stage priority.** The high-amplitude component $k = 8$ exhibits smooth, monotonic error decay from the earliest epochs onward, whereas the low-amplitude components $k = 1$ and $k = 3$ oscillate substantially before eventually converging.
- **Robustness across seeds.** The training trajectory of $k = 8$ maintains small variance throughout optimization, while $k = 1$ and $k = 3$ display large variance, indicating high sensitivity to initialization.

C.1 Ablation on Circuit Depth

We sweep the circuit depth $L \in \{8, 12, 16, 20\}$ while fixing the remaining configuration to the baseline used in Sec. 3. As shown in Fig. 12, at every tested depth the high-amplitude component $k = 8$ decays smoothly and monotonically with small seed variance, whereas the low-amplitude components $k = 1$ and $k = 3$ display markedly non-monotonic trajectories and large variance. Even at the shallowest depth $L = 8$, where the trainable expressivity of the ansatz is limited, the amplitude-induced convergence ordering remains clearly visible. Greater circuit depth uniformly accelerates convergence, but does not alter the qualitative ordering among the three frequencies.

C.2 Ablation on Qubit Count

We vary the qubit count $n \in \{2, 4, 8\}$ while fixing $L = 20$. As illustrated in Fig. 13, the high-amplitude component $k = 8$ achieves rapid and monotonic decay across all qubit counts, whereas the low-amplitude components $k = 1$ and $k = 3$ initially drift upward or oscillate before eventually decreasing. Increasing the qubit count improves the overall convergence speed, but the amplitude-driven ordering and the variance gap between $k = 8$ and $k \in \{1, 3\}$ remain intact.

C.3 Ablation on Data Encoding Strategy

We compare three encoding schemes: $R_Y(x)$ (the baseline), $R_Y(x/2)$, and the composite $R_Y(x) + R_X(x)$. As shown in Fig. 14, under all three encodings $k = 8$ is the only frequency exhibiting a strictly monotonic trajectory with small variance, while $k = 1$ and $k = 3$ remain oscillatory and

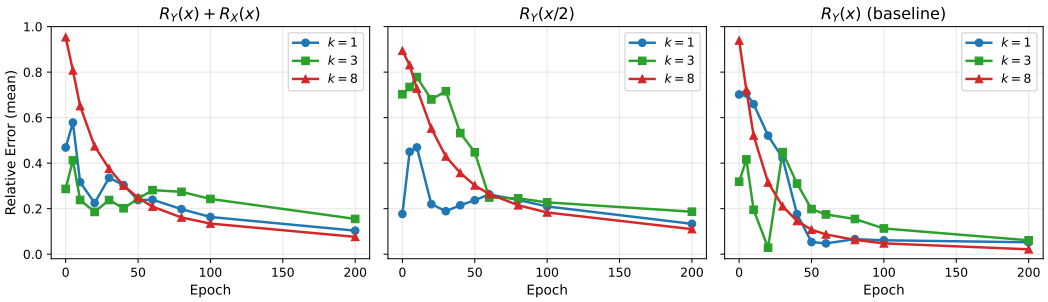


Figure 14: **Ablation on data encoding strategy.** Mean relative error over training epochs for the three encoding schemes. The high-amplitude component $k = 8$ remains the fastest-converging and most stable trajectory under every encoding.

seed-sensitive throughout training. These observations confirm that the spectral neutrality of $\partial \hat{f} / \partial \boldsymbol{\theta}$ established in Sec. 2.1.1 is encoding-independent.

C.4 Summary

Across the 10 configurations spanning four circuit depths, three qubit counts, and three encoding strategies, the conclusion is unanimous: the highest-amplitude component converges first, regardless of its frequency index, and exhibits the smallest variance across random seeds. These ablations collectively provide strong numerical support for Spectral Amplitude Priority as an intrinsic property of QNN training dynamics, decoupled from any specific architectural prior.

D Robustness of Spectral Bias Across Activation Functions

The comparison between QNNs and DNNs in Sec. 3 adopts $\tanh$ as the activation function for the classical baseline. Since prior work has reported that different activation functions (e.g., ReLU) may exhibit slightly different spectral behaviors, one might wonder whether the high-frequency learning failure observed for the DNN baseline is an artifact of this particular choice rather than a structural limitation of classical fully-connected architectures. To rule out this possibility, we perform an additional ablation on the high-frequency-dominated target

$$f_H(x) = 0.1 \sin(x) + 0.1 \sin(3x) + 0.9 \sin(8x),$$

in which we vary the activation function while keeping the depth, width, optimizer, learning rate, and training budget identical to those of the classical baseline used in Sec. 3.

Setup. We evaluate four representative activation functions covering a broad range of inductive biases: (i) $\tanh$, the smooth bounded activation used in our main experiments; (ii) ReLU, the canonical piecewise-linear activation known to induce a kernel with different spectral decay [27]; (iii) Sigmoid, a smooth saturating activation prone to vanishing gradients; and (iv) hat (triangular) activation, a piecewise-linear activation with compact support that, owing to its local structure, in principle offers stronger high-frequency fitting capability. For each configuration we report the mean and standard deviation over multiple independent random seeds.

Results. Table 5 summarizes the training and validation performance of the four activation variants. Although the activation functions induce visibly different optimization behaviors on the training set—with hat achieving the highest training accuracy due to its strong local fitting capacity, while Sigmoid almost fails to learn owing to severe gradient saturation—none of the four DNN variants achieves satisfactory generalization on the high-frequency target. The best validation accuracy across all activations remains around 0.73, which is substantially lower than that attained by the QNN with two orders of magnitude fewer parameters (cf. Fig. 6).

Discussion. A particularly informative case is the hat activation, whose piecewise-linear local support endows the network with strong high-frequency expressive capacity in principle. Indeed, hat attains the highest training accuracy (0.978), indicating that the network is capable of memorizing the high-frequency target. Nevertheless, its validation accuracy (0.663) collapses well below that of

Table 5: Performance of DNNs with different activation functions on the high-frequency target $f_H(x)$. Reported as mean $\pm$ std across multiple random seeds.

Activation	Train Loss	Val Loss	Train Acc.	Val Acc.
$\tanh$	0.1170 ± 0.0931	0.2734 ± 0.0641	0.8830 ± 0.0931	0.7266 ± 0.0641
ReLU	0.1682 ± 0.0107	0.2767 ± 0.0662	0.8318 ± 0.0107	0.7233 ± 0.0662
Sigmoid	0.4151 ± 0.0000	0.4168 ± 0.0000	0.5849 ± 0.0000	0.5832 ± 0.0000
hat	0.0219 ± 0.0156	0.3374 ± 0.0802	0.9781 ± 0.0156	0.6626 ± 0.0802

$\tanh$ and ReLU, exhibiting a pronounced generalization gap that is consistent with the spectral-bias literature: even when high-frequency components can be fit on training points, the gradient-descent dynamics of classical DNNs preferentially align with low-frequency modes during early training, leaving the high-frequency content under-regularized and prone to overfitting.

These observations collectively demonstrate that the inability of DNNs to generalize on high-frequency-dominant targets is a structural and architecture-wide property of classical fully-connected networks, rather than an artifact of the $\tanh$ activation adopted in the main text. This further reinforces the central message of our work: the Spectral Amplitude Priority exhibited by QNNs is a qualitatively distinct training dynamics that classical DNNs do not possess regardless of their activation function.

E Extended Discussion on Spectral-Bias-Mitigating Baselines

This appendix complements Sec. 3.1.3 by providing a more detailed mechanistic discussion of the three classical architectures—FFN [36], SIREN [37], and KAN [38]—against which we benchmark the QNN. All experiments adopt the high-frequency-dominated target function $f_H(x)$ defined in Eq. 8, with each baseline configured to match the QNN parameter budget (≈ 160 parameters) and trained over multiple random seeds. The results visualized in Fig. 8 reveal a unified mechanistic picture across the three architectures, which we discuss in turn.

Fourier Feature Networks. The FFN augments a standard MLP by lifting the input through a random Fourier mapping $\gamma(x) = [\cos(2\pi Bx), \sin(2\pi Bx)]$ with $B \sim \mathcal{N}(0, \sigma^2)$. The scale σ explicitly controls the frequency support of the lifted representation, effectively reshaping the network's neural tangent kernel into a stationary kernel whose bandwidth is governed by σ. Consequently, the model's ability to fit high-frequency targets is gated entirely by this hyperparameter. As shown in Fig. 8(a), small σ values starve the network of high-frequency components, while large σ values introduce spurious high-frequency modes that cause aliasing on the test set, manifesting as the characteristic "U-shape" with widening variance at large σ.

SIREN. SIREN replaces standard activations with sinusoidal nonlinearities of the form $\sin(\omega_0 Wx + b)$, where the scaling factor ω_0 directly modulates the effective frequency support of every hidden layer. This construction grants SIREN strong representational capacity for oscillatory signals provided that ω_0 is sufficiently large. Fig. 8(b) reveals a sharp phase transition: for $\omega_0 \leq 1.5$ the network remains in a low-frequency regime and effectively ignores the dominant high-frequency component of $f_H(x)$; only after ω_0 crosses a critical threshold does the model recover the target. This behavior demonstrates that SIREN does not overcome spectral bias through training dynamics; rather, its hypothesis class itself must be re-centered into the correct spectral band by an externally specified hyperparameter.

KAN. KAN replaces fixed activation functions with learnable B-spline functions, where the spline grid size determines the local resolution of the approximator. The model's expressivity for high-frequency content thus scales with grid resolution: a coarse grid acts as an effective low-pass filter and is structurally incapable of representing the rapid oscillations of $f_H(x)$, whereas finer grids progressively unlock higher-frequency representational capacity, as evidenced by Fig. 8(c). Although KAN's adaptive basis offers strong fitting power once the grid is sufficiently refined, this efficacy is again contingent on a target-dependent capacity hyperparameter, rather than on an intrinsic preference of its training dynamics for high-amplitude spectral components.

Common mechanism and contrast with the QNN. Despite their architectural diversity, FFN, SIREN, and KAN share a common operating principle: each embeds high-frequency expressivity

into the architecture itself, parameterized by a frequency scale (σ, ω_0) or resolution (grid_size) that must be tuned to the target. None of these baselines reorder the optimization preferences of gradient descent over different frequency components; rather, they reshape the hypothesis class so that the conventional low-frequency-first dynamics happen to lie in the correct spectral band. The QNN's behavior is fundamentally different in kind: its preference for high-amplitude spectral components is generated by the eigen-structure of the QNTK and is dictated solely by the target spectrum, with no architectural hyperparameter tied to a particular frequency. Consequently, the QNN attains a stable test MSE around the red dashed line in Fig. 8 across all three subplots, whereas each classical baseline traces out a wide, hyperparameter-dependent envelope. This contrast substantiates our central claim that Spectral Amplitude Priority is an intrinsic dynamical property of QNN optimization, qualitatively distinct from the architecture-driven remedies offered by FFN, SIREN, and KAN.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:The abstract and introduction claim four contributions (Spectral Amplitude Priority principle, expressiveness advantages, residual dynamics via QNTK, and verification on quantum-advantage tasks), all of which are formally derived in Sec.2 and empirically validated in Sec.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the "Remarks and Limitations" paragraph at the end of Sec.1, which explicitly discusses two key limitations: (1) reliance on ideal circuit simulations without hardware noise, and (2) the theoretical derivation being restricted to gradient-based optimization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The two key assumptions—(1) the lazy training regime (sufficiently small learning rate ensuring near-constant QNTK) and (2) the ideal circuit model—are explicitly stated in Sec.2. The lazy-training assumption is further empirically verified in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies the QNN architecture (qubits, depth, encoding scheme), target functions, learning rate (η =0.01), parameter initialization range ([0, 2π]), DNN baseline configurations, software framework (PennyLane), and hardware (i7-14700HX CPU, RTX 4060 GPU). Ablation settings are also fully detailed in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code and data are not included in the current submission. We plan to release the implementation code upon acceptance to facilitate reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper reports circuit configurations (qubits, depth, encoding), learning rate, initialization range, target functions, DNN architectures, activation functions, loss functions, and training epochs. Ablation hyperparameters are detailed in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 5 reports mean ± std over multiple random seeds, Fig. 8 shows shaded one-std regions across seeds, and the ablation studies average over multiple independent seeds with variance discussion. The variability source (random parameter initialization) is stated.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the hardware platform (Intel Core i7-14700HX CPU, NVIDIA GeForce RTX 4060 GPU). The experiments are small-scale quantum circuit simulations (≤ 8 qubits) with modest computational requirements.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research is foundational theoretical and simulation work on quantum neural network training dynamics. It does not involve human subjects, sensitive data, or dual-use concerns, and conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper is foundational theoretical research on understanding QNN training dynamics. It has no direct societal impact or foreseeable negative applications

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper presents theoretical analysis and small-scale quantum circuit simulations. It does not release pre-trained models, datasets, or any artifacts that pose risks for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: We cite the original papers for all software and datasets used. All assets used are publicly available open-source tools or standard benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This paper does not release new assets. Our contribution is a theoretical analysis and empirical validation of the Spectral Amplitude Principle, not a new dataset, model, or code package.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects. All experiments are computational simulations of quantum circuits.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects. Our work is purely computational and theoretical.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.