

The PID Controller Strikes Back: Classical Controller Helps Mitigate Barren Plateaus in Noisy Variational Quantum Circuits

Zhehao Yi ZHEHAO.YI@UAH.EDU and **Rahul Bhadani** RAHUL.BHADANI@UAH.EDU
AI, Autonomy, Resilience, Control (AARC) Lab, Electrical & Computer Engineering, The University of Alabama in Huntsville, AL, 35899, USA

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Variational quantum algorithms (VQAs) combine the advantages of classical optimization and quantum computation, making them one of the most promising approaches in the Noisy Intermediate-Scale Quantum (NISQ) era. However, when optimized using gradient descent, VQAs often suffer from the vanishing gradient problem, commonly known as the barren plateau. Various methods have been proposed to mitigate this issue. In this work, we propose a hybrid approach that integrates a classical proportional-integral-derivative (PID) controller with a neural network to update the parameters of variational quantum circuits. We refer to this method as NPID, which aims to mitigate the barren plateau. The proposed algorithm is tested on randomly generated quantum input states and random quantum circuits with parametric noise to evaluate its universality, and additional simulations are conducted under different noise rates to examine its robustness. The effectiveness of the proposed method is evaluated based on its convergence speed toward the target cost value. Simulation results show that NPID achieves a convergence efficiency 2–9 times higher than Neural Enhanced Quantum Parametric Model (NEQP) and Standard Quantum Vanilla Model (QV), with performance fluctuations averaging only 4.45% across different noise levels. These results highlight the potential of integrating classical control theory into quantum optimization, providing a new perspective for improving the trainability and stability of variational quantum algorithms.

Keywords: Variational Quantum Circuits, Barren Plateau, PID Controller, Neural Networks

1. Introduction

Quantum computing has emerged as one of the most promising perspectives in modern computational science (Gill et al., 2022; Meier and Yamasaki, 2025; Ramezani et al., 2020; Nielsen and Chuang, 2010). Rooted in the principles of quantum mechanics and harnessing the phenomena of entanglement and superposition, quantum computing can perform certain classes of computations more efficiently than its classical counterpart and can represent exponentially richer information states. In quantum computation, the fundamental unit of information is the qubit, which encodes quantum information. With rapid advances in quantum machine learning (QML) and quantum information processing, quantum computing is increasingly being explored across diverse domains such as transportation, healthcare, and finance (Herman et al., 2023; Tayur and Tenneti, 2024; Karahalios et al., 2025).

Quantum machine learning is commonly implemented through variational quantum algorithms (VQAs) (Cerezo et al., 2021a). These algorithms typically consist of four key components: data encoding, parameterized quantum circuits, measurements, and a cost function. The quantum components include the encoding, circuit, and measurement processes, while the cost function represents the classical optimization part. This hybrid structure closely resembles that of traditional neural

networks, where the cost function guides the parameter updates throughout the model. Owing to this flexible architecture, VQAs have become one of the most promising frameworks for quantum machine learning in the Noisy Intermediate-Scale Quantum (NISQ) era (Lau et al., 2022).

Currently, a significant portion of research in quantum machine learning relies on VQAs. In the transportation domain, researchers have employed variational quantum circuits (VQCs) to construct Quantum Deep Convolutional Neural Networks (QDCNNs) for modeling shadow image processing under complex environmental conditions (Meghanath et al., 2025), achieving promising performance. Other studies have proposed the UU^\dagger algorithm for quantum image processing, demonstrating robustness and strong resistance to noise (Innan et al., 2025). A hybrid quantum genetic algorithm (HQGA) has also been developed for structural damage identification, improving detection efficiency while maintaining stability (Xu et al., 2025). In addition, optimization-oriented approaches such as the Quantum Approximate Optimization Algorithm (QAOA) (Zhou et al., 2023) and the Quantum Alternating Direction Method of Multipliers (QADMM) (Nie et al., 2025) have been explored for target optimization and distributed quantum computing, respectively.

Despite a large number of advantages of VQAs, they suffer from a critical problem known as the barren plateau (Larocca et al., 2025; McClean et al., 2018). This issue arises when the number of qubits or the depth of the quantum circuit increases. As a result, the gradient used to update model parameters vanishes exponentially. The optimization process often becomes trapped in local optima or stalls entirely. This can lead to significant computational overhead without any meaningful improvement in model performance. The barren plateau effect is influenced by various factors, including circuit initialization, depth, cost function design, and quantum noise. To address this problem, several mitigation strategies have been proposed. These include tailored cost function design (Cerezo et al., 2021b), layerwise training (Skolik et al., 2021), improved parameter initialization (Kashif et al., 2024), neural network generation of quantum states (Yi and Bhadani, 2025), and the use of neural-network-based optimizers for parameter updates (Yi et al., 2025; Friedrich and Maziero, 2022).

In this paper, we propose an alternative approach to mitigate the barren plateau problem. We employ a neural proportional–integral–derivative (NPID) controller to update the parameters of a variational quantum algorithm and find that this classical control mechanism effectively enhances the performance of the VQAs under noise.

This article is organized as follows. Section 2 provides the necessary background information. Section 3 describes the proposed methodology. Section 4 presents and analyzes the experimental results, and Section 5 concludes the paper with a summary and future outlook.

2. Background

2.1. Variational Quantum Algorithms

Variational quantum algorithms are hybrid quantum–classical frameworks that optimize parameterized quantum circuits to minimize a given cost function. A VQA consists of four important components: encoding, quantum operation, measurement, and classical optimization. In the encoding stage, classical data are transformed into quantum states by using angle, basis, or amplitude encoding to serve as inputs to the circuit (Rath and Date, 2024), as shown in Figure 1.

The quantum circuit is composed of parameterized single-qubit rotation gates (e.g., R_x , R_y , R_z) and multi-qubit entangling gates (e.g., $CNOT$, CZ), which together define a tunable operation on the quantum state (Wong, 2023; Williams, 2011; DiVincenzo, 1998). During this transformation

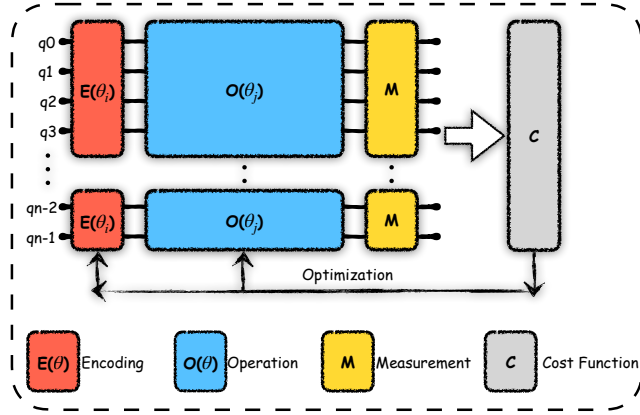


Figure 1: The construction of the Variational Quantum Algorithm.

process, intermediate measurements are prohibited because they would collapse the superposition, halting further computation.

After the operation completes, measurement yields a probability distribution over computational basis states. These probabilities are used to evaluate the cost function, whose gradient information is used in the classical parameter optimization. While VQAs have demonstrated strong expressivity and adaptability across quantum machine learning tasks, they also suffer from the barren plateau problem, in which gradients vanish exponentially as the number of qubits or circuit depth increases.

2.2. Barren Plateau

Barren plateaus are among the major challenges hindering the scalability of VQAs (Friedrich and Maziero, 2022; McClean et al., 2018). Since VQAs typically rely on gradient-based optimization to update circuit parameters, the occurrence of a barren plateau causes the gradients to vanish, which, as they approach zero, impedes the training process. Let the parameterized quantum circuit be represented by $U(\theta_i)$, the input quantum state is $|\psi_{in}\rangle$, then the output quantum state is:

$$|\psi_{out}\rangle = U(\theta_i) |\psi_{in}\rangle \quad (1)$$

Given a cost function \mathcal{L} , the gradient of \mathcal{L} with respect to each trainable parameters θ_i , can be expressed as:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{\partial \langle \psi_{out} | \hat{M} | \psi_{out} \rangle}{\partial \theta_i} \quad (2)$$

where the \hat{M} denotes the measurement consequences (Cerezo et al., 2021a).

As shown in previous analyses, in deep quantum circuits, highly entangled circuits or those containing a large number of qubits, the gradient converges exponentially toward zero, as expressed in Equation (3),

$$\frac{\partial \mathcal{L}}{\partial \theta_i} \leq G(n), G(n) \in O\left(\frac{1}{a^n}\right), a \geq 1 \quad (3)$$

rendering the optimization ineffective and resulting in the barren plateau.

2.3. Proportional–integral–derivative Controller

Proportional–integral–derivative controllers have been widely employed in industrial and engineering systems for decades (Åström and Hägglund, 2006). In a closed-loop linear control system, the controller regulates the system output to track a desired reference. The error signal $e(t)$, defined as the difference between the reference and the actual output, serves as the basis for adjustment. The PID control law comprises three components: Proportional control multiplies the error by a proportional gain K_p :

$$P = K_p e(t) \quad (4)$$

Integral control integrates the error over time and is scaled by an integral K_i :

$$I = K_i \int_0^t e(\tau) d\tau \quad (5)$$

and Derivative control differentiates the error with respect to time and multiplies it by a derivative gain K_d :

$$D = K_d \frac{de(t)}{dt} \quad (6)$$

By combining these three terms, the PID controller uses current, accumulated, and predicted error information to tune the control system, thereby enhancing system stability. However, specific control modes can be realized by setting one or more of the gains to zero, resulting in P, PI, or PD controllers.

3. Method

In this section, we present the proposed methodology. Since the PID controller is fundamentally designed for linear systems (Åström and Hägglund, 2006), we first demonstrate in Appendix B that a parameterized quantum circuit can be regarded as a linear system under certain conditions.

To ensure the generality of the experiments, random quantum input states were employed. These states were generated by sequentially applying quantum rotation gates to the ground state $|0\rangle$, shown in Equation (7), with the rotation parameters randomly initialized to guarantee the randomness of the input states.

$$|\psi_{in}\rangle = \bigotimes_1^n \{R_z(\theta_k)R_y(\theta_j)R_x(\theta_i) |0\rangle\} \quad (7)$$

We constructed random quantum circuits with circuit depth D to evaluate the universality of our proposed method. Each layer begin with single-qubit rotation gates applied to all qubits. Qubits were then randomly paired and entangled using CNOT gates, followed by additional R_x and R_z rotation gats to increase complexity, as shown in Figure 2. Because true randomness cannot be achieved computationally, we maximized the circuits complexity to approximate the behavior of truly random quantum circuits. Given that practical computation is never perfectly noise-free, we introduced parameter noise to all parameterized gates. For any parameters set θ , a random perturbation Δ was added; δ represents the noise rate, and α is a random parameter set of the same as θ , as shown in the following Equation (8).

$$\begin{aligned} \hat{\theta} &= \theta + \Delta \\ \Delta &= \delta \cdot \alpha \end{aligned} \quad (8)$$

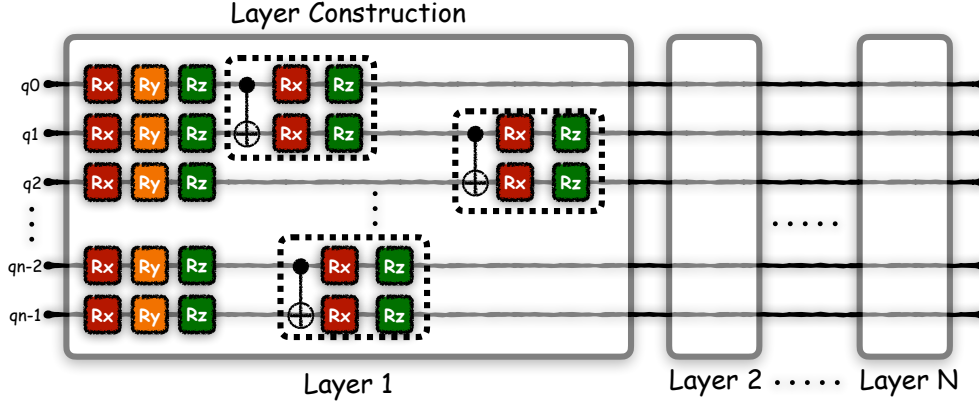


Figure 2: The Construction of the random quantum circuit. The gray boxes represent individual circuit layers, while the black dashed boxes indicate the gates applied between pairs of randomly selected qubits.

Next, we employed the cost function in Equation (9) to compute the loss value, consistent with that used in previous studies (Friedrich and Maziero, 2022):

$$\mathcal{L} = 1 - \frac{1}{n} \sum_{i=1}^n \text{Tr}\{|0\rangle\langle 0|_i \otimes I_i |\psi_{in}\rangle\} \quad (9)$$

In this formulation, the i -th qubit interacts with $|\psi_{in}\rangle$, while I_i interacts with all other qubits except i . This formula represents one minus the probability that each qubit in the output quantum state is in the ground state $|0\rangle$. Minimizing this cost function requires the latter term to approach 1, which in turn enforces the number of output qubits to be close to $|0\rangle^{\otimes n}$.

After obtaining the result from the cost function $e = \mathcal{L}$, the PID control process begins. First, the three components of the PID controller are computed. In the proportional control term, P_e is defined as the current output loss e . In the integral control term, I_e is defined as the sum of the current and previous loss e_{pre} , approximating the integration over time. In the derivative control term, D_e is defined as the difference between the current and previous loss. Second, the three PID coefficients are determined. Unlike traditional PID control, where the coefficients are fixed, we introduce a small fully connected neural network that dynamically adjusts these coefficients during control. This neural network receives P_e , I_e , D_e and e as input and the outputs the three adaptive coefficients K_p , K_i , and P_d . Next, the corresponding terms are multiplied and summed to obtain the PID output O_{pid} . The gradient with respect to parameters set $\frac{\partial \mathcal{L}}{\partial \theta}$ is then computed, and the parameter update process for the list θ is expressed as Equation (10), where $\hat{\theta}$ means the noisy parameter list, and lr means the learning rate.

$$\hat{\theta}_{i+1} = \hat{\theta}_i - lr \cdot \frac{\partial \mathcal{L}}{\partial \theta} \cdot O_{pid} \quad (10)$$

The algorithm is as follows:

The overall training flowchart is shown in Figure 3. NPID denotes the Neural PID Model, NEQP represents the Neural Enhanced Quantum Parametric Model, and QV refers to the Standard Quantum Vanilla Model. In NEQP, the neural network receives a random parameter set as input and outputs a parameter vector matching the required circuit dimensionality (Friedrich and Maziero,

Algorithm 1 Calculate update parameters set $\hat{\theta}_{i+1}$

Require: e is the current loss, e_{pre} is the previous loss, $\frac{\partial \mathcal{L}}{\partial \theta}$ is the gradient, lr means the learning rate, θ_i means the previous parameters set, and ϕ denotes the neural network.

- 1: $P_e = e$
 - 2: $I_e = e + e_{pre}$
 - 3: $D_e = e - e_{pre}$
 - 4: $K_p, K_i, K_d = \phi(e, P_e, I_e, D_e)$
 - 5: $O_{pid} = K_p \cdot P_e + K_i \cdot I_e + K_d \cdot D_e$
 - 6: $\hat{\theta}_{i+1} = \theta_i - lr \cdot \frac{\partial \mathcal{L}}{\partial \theta} \cdot O_{pid}$
-

2022; Yi et al., 2025). These parameters are then sequentially applied to the quantum circuit to generate the output state. The cost function is computed based on this output state, and the resulting loss value is used to update the weights w and biases b of the neural network, rather than the input parameters themselves. In NPID, the computed loss value e serves two purposes: it is used in the PID control computation to update the parameter list θ of the quantum circuit, and simultaneously drives the gradient updates of the neural network that generates the PID gains. In contrast, QV directly generates the required parameter vector and updates it through the same gradient-based optimization steps.

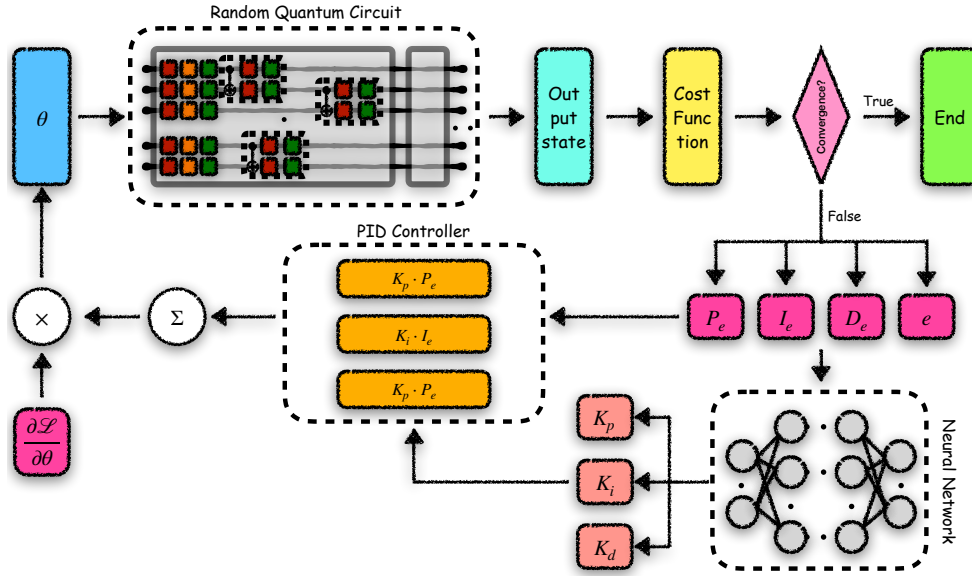


Figure 3: The construction of the Neural Proportional–Integral–Derivative Model.

The architectures of the neural networks used in the simulations are shown in the Table 1. NPID and NEQP share the same number of layers, while NPID and NEQP-S have an identical number of neurons per layer except for the final output, and NEQP-L employs a larger number of neurons in each layer. A Softplus activation function is added at the output of the NPID network, ensuring that the PID controller gains remain positive, as is typical in control systems (Ogata, 2010; Dorf

and Bishop, 2016). The goal of maintaining an identical number of layers while varying the neuron count enables an evaluation of the proposed model’s learning efficiency (Lu et al., 2017).

Table 1: The construction of neural networks

NPID	NEQP-S	NEQP-L
Linear(4, 32)	Linear(4, 32)	Linear(32, 256)
Tanh()	Tanh()	Tanh()
Linear(32, 64)	Linear(32, 64)	Linear(256, 256)
Tanh()	Tanh()	Tanh()
Linear(64, 3)	Linear(64, θ^*)	Linear(256, θ^*)
Softplus()		

* The parameters list for the quantum circuit.

In summary, we propose a hybrid classical control system with a quantum variational framework to mitigate the barren plateau. In the following section, we will present our simulation results.

4. Results

We first describe the experimental setup. The number of qubits ranges from 7 to 12. The depth of each random quantum circuit is set to $D = n^2 \log n$, where n denotes the number of qubits, to ensure the circuits are sufficiently complex to induce the barren plateau phenomenon. For each of the four models, we conduct five independent runs with distinct circuit initializations and parameter seeds. All models are optimized using stochastic gradient descent (SGD) optimizer. The training objective is to reduce the cost function value \mathcal{L} below 0.001, and the noise rate is 0.01. All models are developed in PyTorch (Paszke et al., 2019), and the quantum circuits are constructed with TorchQuantum (Wang et al., 2022). The experiments are executed on a workstation equipped with an AMD Ryzen 9 7960X 24-core processor and an NVIDIA RTX 4090 GPU with 24 GB of memory.

Figure 4 illustrates the relationship between the number of iterations and the loss value. In this experiment, the maximum number of iterations was set to 1500. As shown in the figure, as the number of qubits increases, NEQP-L consistently requires the full number of iterations to converge, while the iterations needed by NEQP-S and QV also grow rapidly with system size. In contrast, NPID maintains a consistently fast decrease in loss. Although its iteration count increases slightly with the number of qubits, it remains significantly lower than that of the other two models. This demonstrates that NPID effectively mitigates the barren plateau effect associated with increasing qubit count and circuit depth. The shaded regions represent the mean and variance of the recorded loss values, showing that NPID exhibits stable convergence behavior. In contrast, QV and NEQP-S perform worse as the number of qubits increases, while NEQP-S and NEQP-L display pronounced fluctuations under noisy quantum circuits and remain trapped in a barren plateau. These results also indicate that simply increasing the number of neurons in a neural network does not necessarily improve its performance. Although NEQP-S performs well for smaller qubits, it, like NEQP-L, eventually encounters the barren plateau as the system size increases.

Figure 5 illustrates the average number of iterations required by each model across different qubit counts. With the maximum iteration limit set to 1500, both QV and NEQP-L reached this upper bound once the number of qubits exceeded nine, while NEQP-S performed slightly better but

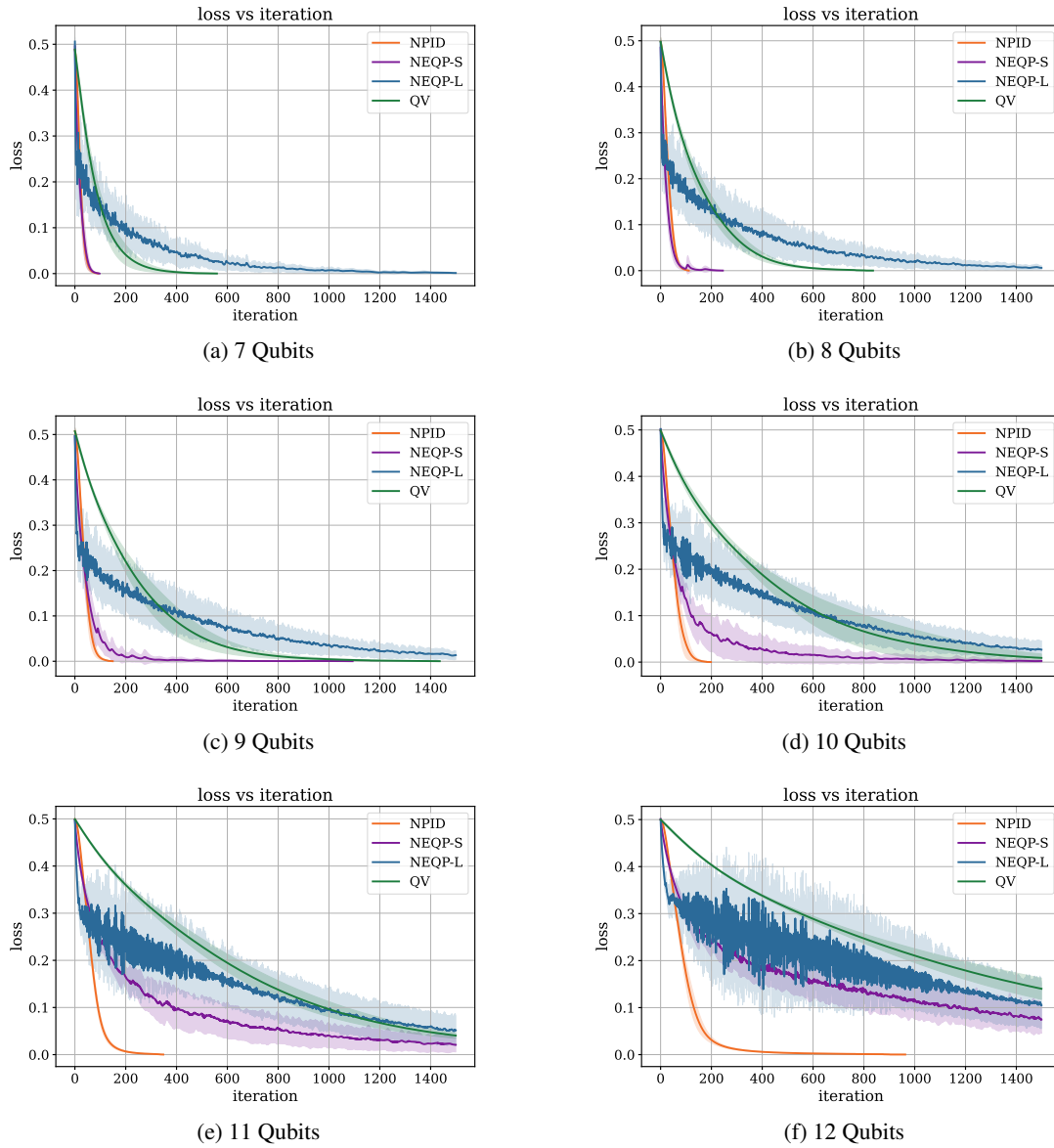


Figure 4: Average loss versus number of iterations. The shaded regions represent the mean \pm standard deviation over five independent runs.

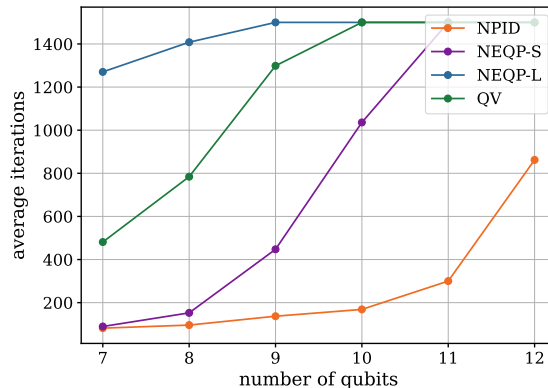


Figure 5: Number of iterations needed for the target cost function for different numbers of qubits.

still grew rapidly. In contrast, the NPID model showed superior performance: although its iteration count increased slightly with the number of qubits, its overall convergence speed remained significantly higher than that of NEQP-S, NEQP-L, and QV. Table 2 summarizes the average iteration count and convergence efficiency for each configuration. The average convergence efficiency \hat{E}_c is computed as $avg(\sum \frac{max(Iteration)}{conv(Iteration)})$, where $max(Iteration)$ denotes the maximum number of iterations and $conv(Iteration)$ denotes the iterations required for convergence. As observed, NPID achieves an average convergence efficiency of 11.908, while the corresponding values for NEQP-S, NEQP-L, and QV are 5.544, 1.249, and 1.837, respectively. This demonstrates that NPID converges approximately 2–9 times more efficiently than the other models, achieving faster convergence and effectively mitigating the barren plateau.

Table 2: Average Number of iterations required to reach the target cost function value for different numbers of qubits.

Model	7 Qubits	8 Qubits	9 Qubits	10 Qubits	11 Qubits	12 Qubits	\hat{E}_c
NPID	82	96	138	177	328	881	11.908
NEQP-S	90	153	448	1036	1500	1500	5.544
NEQP-L	1270	1408	1500	1500	1500	1500	1.249
QV	481	785	1299	1500	1500	1500	1.837

Table 3 compares the number of iterations required by the NPID model under different noise rates. Parameter noise was introduced at rates of 0.03, 0.05, 0.07, and 0.09, and the corresponding iteration counts for convergence were recorded for the same qubit configurations, as shown in Figure 6. The results show that, for a fixed number of qubits, the iteration counts remain nearly consistent, fluctuating by only about 2–7% (4.45% on average), calculated as the ratio of standard deviation to mean. This indicates that, within a physically realistic range of parameter noise, variations in noise have little impact on the upper bound of the model’s convergence complexity. These findings further demonstrate the strong noise robustness of the proposed NPID model, suggesting its ability to effectively mitigate the barren plateau even in quantum circuits affected by parameter perturbations.

5. Summary

In this study, we propose an algorithm that incorporates a classical PID controller to update parameters in variational quantum circuits, it is an alternative optimization scheme in traditional gradient

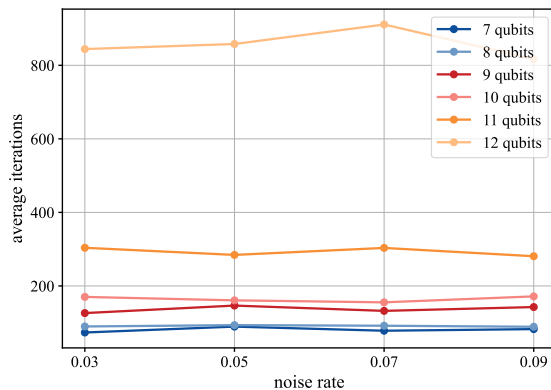


Figure 6: Average number of iterations required for NPID under different noise rates.

Table 3: Average Iterations comparison of NPID under varying qubit numbers and Noise rates.

Noise Rate	7 Qubits	8 Qubits	9 Qubits	10 Qubits	11 Qubits	12 Qubits
0.03	74	90	127	171	304	845
0.05	90	94	147	161	285	858
0.07	79	92	133	156	304	911
0.09	83	89	143	172	281	817
Fluctuation Rate	7.18%	2.10%	5.76%	4.08%	3.61%	3.98%

descent. Unlike Normalized Gradient Descent (NGD) (Suzuki et al., 2021) which depends on the present gradient re-scaling (normalization), our NPID framework introduces an adaptive control unit that uses historical loss value to correct the optimization trend that leads to faster convergence. We first outline the basic structure and workflow of a variational quantum algorithm, define the barren plateau phenomenon, and review several existing mitigation strategies, along with a brief construction of the PID controller. Next, we describe our proposed approach, which demonstrates universality by employing randomly generated qubit input states and randomly constructed quantum circuits. We then illustrate how the NPID model updates the circuit parameters during training. Simulation results show that the proposed NPID achieves faster and more stable convergence compared to NEQP and QV, without exhibiting significant fluctuations. Moreover, it maintains consistent performance across various noise levels, demonstrating strong robustness and scalability. This study bridges classical control theory and quantum optimization, offering a new perspective for enhancing the trainability of quantum neural networks in the NISQ era. Future work will explore extending this approach to other classical control strategies for mitigating the barren plateau in quantum machine learning with different types of noise, such as Pauli Channel and Damping Channel, and evaluating the scalability of the approach across larger-scale quantum systems.

Acknowledgments

This work was supported by a Graduate Research Assistantship at the AARC Lab, Department of Electrical and Computer Engineering, University of Alabama in Huntsville. The authors would like to thank Md Saiful Islam and Sophia Vanderwaal for some clarifications on issues related to control systems and physics, respectively.

Code and Artifacts Availability: All the artifacts and data used to produce this research are available at https://github.com/AARC-lab/L4DC26_VQA.

References

- K. J. Åström and T. Hägglund. *PID Control*. IEEE Control Systems Magazine, 2006.
- M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021a.
- M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, 2021b.
- D. P. DiVincenzo. Quantum gates and circuits. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):261–276, 1998.
- Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. Pearson Education, 2016.
- L. Friedrich and J. Maziero. Avoiding barren plateaus with classical deep neural networks. *Physical Review A*, 106(4):042433, 2022.
- S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya. Quantum computing: A taxonomy, systematic review and future directions. *Software: Practice and Experience*, 52(1):66–114, 2022.
- D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev. Quantum computing for finance. *Nature Reviews Physics*, 5(8):450–465, 2023.
- N. Innan, B. K. Behera, S. Al-Kuwari, and A. Farouk. Qnn-vrcs: A quantum neural network for vehicle road cooperation systems. *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- A. Karahalios, S. Tayur, A. Tenneti, A. Pashapour, F. S. Salman, and B. Yıldız. A quantum-inspired bilevel optimization algorithm for the first responder network design problem. *INFORMS Journal on Computing*, 37(1):172–188, 2025.
- M. Kashif, M. Rashid, S. Al-Kuwari, and M. Shafique. Alleviating barren plateaus in parameterized quantum machine learning circuits: Investigating advanced parameter initialization strategies. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2024.
- M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo. Barren plateaus in variational quantum computing. *Nature Reviews Physics*, pages 1–16, 2025.
- J. W. Z. Lau, K. H. Lim, H. Shrotriya, and L. C. Kwek. Nisq computing: where are we and where do we go? *AAPPS Bulletin*, 32(1):27, 2022.
- Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

- J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.
- A. Meghanath, S. Das, B. K. Behera, M. A. Khan, S. Al-Kuwari, and A. Farouk. Qdcnn: Quantum deep learning for enhancing safety and reliability in autonomous transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- F. Meier and H. Yamasaki. Energy-consumption advantage of quantum computation. *PRX Energy*, 4(2):023008, 2025.
- H. Nie, D. An, and Z. Wen. Quantum alternating direction method of multipliers for semidefinite programming. *arXiv preprint arXiv:2510.10056*, 2025.
- M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.
- Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 2010.
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi. Machine learning algorithms in quantum computing: A survey. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2020.
- M. Rath and H. Date. Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology*, 11(1):72, 2024.
- A. Skolik, J. R. McClean, M. Mohseni, P. Van Der Smagt, and M. Leib. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence*, 3(1):5, 2021.
- Y. Suzuki, H. Yano, R. Raymond, and N. Yamamoto. Normalized gradient descent for variational quantum algorithms. In *IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 1–9. IEEE, 2021.
- S. Tayur and A. Tenneti. Quantum annealing research at cmu: algorithms, hardware, applications. *Frontiers in Computer Science*, 5:1286860, 2024.
- Hanrui Wang, Yongshan Ding, Jiaqi Gu, Zirui Li, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. Quantumnas: Noise-adaptive search for robust quantum circuits. In *IEEE International Symposium on High-Performance Computer Architecture*, 2022.
- C. P. Williams. Quantum gates. In *Explorations in Quantum Computing*, pages 51–122. Springer London, 2011.
- H. Y. Wong. Quantum gate introduction: Not and cnot gates. In *Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps*, pages 133–141. Cham: Springer International Publishing, 2023.

- L. Xu, X. Wang, Z. Wang, and G. Cao. Hybrid quantum genetic algorithm for structural damage identification. *Computer Methods in Applied Mechanics and Engineering*, 438:117866, 2025.
- Z. Yi and R. Bhadani. Neural-network generated quantum state can mitigate the barren plateau in variational quantum circuits. In *Frontiers in Optics (FiO)*, page FW6D.3. Optica Publishing Group, 2025.
- Z. Yi, Y. Liang, and H. Situ. Enhancing variational quantum circuit training: an improved neural network approach for barren plateau mitigation. *Physica Scripta*, 100(8):086004, 2025.
- Z. Zhou, Y. Du, X. Tian, and D. Tao. Qaoa-in-qaoa: Solving large-scale maxcut problems on small quantum machines. *Physical Review Applied*, 19(2):024027, 2023.

Appendix A. Background on Quantum Computing

In quantum computing, Dirac notation is commonly used to represent quantum states in a Hilbert space (Nielsen and Chuang, 2010). A single qubit pure quantum state can be written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (\text{A.1})$$

where α and β are complex coefficients satisfying $|\alpha|^2 + |\beta|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ form the computational basis and can be represented in matrix form.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (\text{A.2})$$

This expression indicates that a qubit can exist in a superposition of the basis states, where measurement outcomes are obtained probabilistically according to the squared magnitudes of the coefficients. For a n qubits system, the corresponding Hilbert space has dimension of 2^n , and multi qubit states are constructed using tensor products.

Quantum operations are performed by applying quantum gates, which are represented by unitary operators acting on quantum states. Common single qubit gates include the rotation gates

$$R_x(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad (\text{A.3})$$

which rotate the quantum state around different axes on the Bloch sphere. The Hadamard gate is often used to create superposition states

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (\text{A.4})$$

while multi qubit gates such as the CNOT gate are used to control the qubit, where the first qubit is referred to the control qubit and the second qubit is the target qubit, if the control qubit is $|0\rangle$, then leave the target qubit unchanged, otherwise apply a R_x to the target qubit.

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{A.5})$$

Measurement causes the quantum state to collapse onto one of the computational basis states with a certain probability. In practice, outcomes are estimated through expectation values obtained from repeated measurements. The expectation value of an observable M is given by

$$\langle M \rangle = \langle \psi | M | \psi \rangle \quad (\text{A.6})$$

As a result, these expectation values serve as the interface between the quantum circuit and the classical optimization loop. They are used to compute the loss value and update the parameters in variational quantum algorithms.

Appendix B. Proof of linearity of quantum circuit

We begin by briefly reviewing a fundamental concept from quantum mechanics. In quantum theory, any observable or transformation physical quantity O_p can be represented as a linear operator acting on a Hilbert space. For any two quantum states $|\psi_a\rangle$ and $|\psi_b\rangle$, and for any complex scalars a and b , we have

$$O_p(a \cdot |\psi\rangle_a + b \cdot |\psi\rangle_b) = a \cdot O_p |\psi_a\rangle + b \cdot O_p |\psi_b\rangle \quad (\text{B.1})$$

For a quantum circuit represented by the unitary operator U , the entire transformation can be expressed as

$$|\psi_{out}\rangle = U |\psi_{in}\rangle \quad (\text{B.2})$$

Therefore, for any two quantum states $|\psi_m\rangle$ and $|\psi_n\rangle$, and for any complex scalars m and n , we have

$$U(m \cdot |\psi\rangle_m + n \cdot |\psi\rangle_n) = m \cdot U |\psi_m\rangle + n \cdot U |\psi_n\rangle \quad (\text{B.3})$$

Although the unitary operator $U(\theta)$ is generally nonlinear in its parameters θ , under a small-perturbation approximation $\Delta\theta \rightarrow 0$, it can be locally linearized as

$$\begin{aligned} U(\theta) &= e^{-i\theta H} \\ U(\theta + \Delta\theta) &= e^{-i(\theta + \Delta\theta)H} \\ &= e^{-i\theta H} \cdot e^{-i\Delta\theta H} \\ &\approx e^{-i\theta H} \cdot (I - i\Delta\theta H) \\ &= U(\theta) - i\Delta\theta H U(\theta) \end{aligned} \quad (\text{B.4})$$

where H denotes the corresponding Hermitian matrix. This local linearity justifies treating the parameterized quantum circuit as a linear system for the purpose of applying PID control theory.