Evaluating Efficacy of Model Stealing Attacks and Defenses on Quantum Neural Networks

Satwik Kundu
The Pennsylvania State University
State College, Pennsylvania, USA
sxk6259@psu.edu

Debarshi Kundu
The Pennsylvania State University
State College, Pennsylvania, USA
dqk5620@psu.edu

Swaroop Ghosh
The Pennsylvania State University
State College, Pennsylvania, USA
szg212@psu.edu

ABSTRACT

Cloud hosting of quantum machine learning (QML) models exposes them to a range of vulnerabilities, the most significant of which is the model stealing attack. In this study, we assess the efficacy of such attacks in the realm of quantum computing. Our findings revealed that model stealing attacks can produce clone models achieving up to 0.9× and 0.99× clone test accuracy when trained using Top-1 and Top-k labels, respectively (k : num_classes). To defend against these attacks, we propose: 1) hardware variation-induced perturbation (HVIP) and 2) hardware and architecture variation-induced perturbation (HAVIP). Despite limited success with our defense techniques, it has led to an important discovery: QML models trained on noisy hardwares are naturally resistant to perturbation or obfuscation-based defenses or attacks.

ACM Reference Format:
Satwik Kundu, Debarshi Kundu, and Swaroop Ghosh. 2024. Evaluating Efficacy of Model Stealing Attacks and Defenses on Quantum Neural Networks. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3649476.3658806

1 INTRODUCTION

Training QML models is a complex task, intertwining quantum computing's nuances with conventional machine learning. Key challenges involve efficient data encoding, managing qubit constraints for feature representation, and appropriately determining crucial hyperparameters, such as learning rate and QNN structure. Noise in today's NISQ-era quantum devices further complicates QNN training. Given the nascent state of quantum technologies, training a QML model demands significant resources and multiple optimization iterations. The rising necessity for large quantum datasets, coupled with expert knowledge and computational power, underscores the promise of Quantum Machine-Learning-as-a-Service (QMLaaS) [5]. As quantum infrastructure remains intricate and costly, many entities are poised to adopt QMLaaS, enabling them to leverage QNN benefits without direct infrastructural challenges.

While QMLaaS would democratize access to cutting-edge PQC based models, it also exposes QML models to adversarial attacks on cloud platforms. Notably, there's a growing concern over "model stealing" attacks, where attackers can replicate a model's architecture and behavior by systematic querying [4]. Attackers may be motivated to steal models to

(a) avoid paying fees for the services, (b) create synthetic datasets in fields like drug discovery and finance where original datasets are costly using GANs [3], and (c) transition from black-box to white-box attack strategies by accessing the model's internals [6]. This would not only jeopardizes the intellectual property rights of these proprietary PQC-based models but also casts shadows on the security robustness of future QMLaaS platforms. As the quantum computing landscape matures and QNNs gain traction, comprehending and countering the risks of model stealing is of utmost importance. This paper analyzes these potential vulnerabilities to fortify the next wave of QML models.

QNNs differ fundamentally from DNNs due to their reliance on PQCs as their core architectural component. PQCs are inherently sensitive to noise and errors due to fragile nature of quantum states in contrast to DNNs, which are generally robust to errors. The variability in noise across different quantum hardware further complicates the predictability of QNN behavior, especially in the context of model stealing attacks and defenses. The above aspects warrant a fresh look at QNN model stealing attacks.

In this study, i) we assess the effectiveness of model stealing attacks on hybrid QNNs across diverse datasets *for the first time* to the best of our knowledge. When viewing the cloud-based QNN as a black-box, an attacker can iteratively query the victim QNN for inputs, subsequently building a dataset from the returned outputs. Using this acquired dataset, the adversary can then train a clone model that mirrors the performance of the original, stealing it's functionality, ii) capitalize on the inherent noise and architectural variability to propose two perturbation-based defense methods: (a) hardware variation-induced perturbation (HVIP) and (b) hardware and architecture variation-induced perturbation (HAVIP) and, iii) show that training QNNs in noisy conditions actually increases their resistance to perturbation-based defenses, implying a heightened robustness against perturbation-based adversarial attacks as well.

2 PROPOSED ATTACK AND DEFENSE METHODOLOGIES

2.1 Adversary's Knowledge

Here, we assume QNN model hosted on the cloud, treated as a black box. This means the user, acting as an adversary, is unaware of the model's architecture, size (i.e., number of qubits and number of parametric layers), hyperparameters, or the dataset on which it was originally trained. The only knowledge the adversary possesses is the input-output structure; they understand the format of the data

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658806

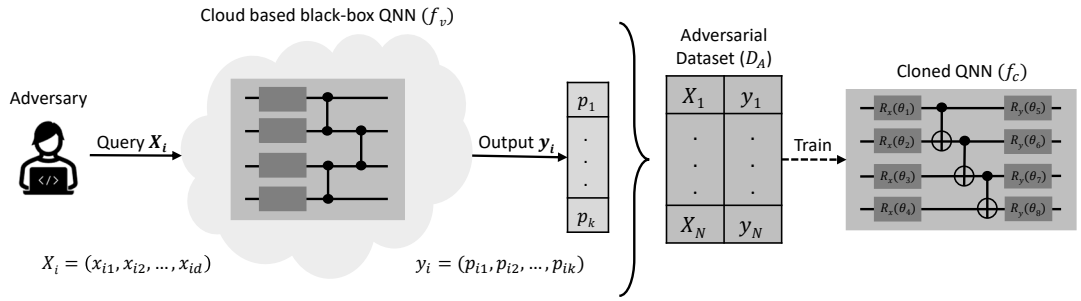

Figure 1: An adversary sends a query vector $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ to a cloud-based victim QNN (f_v), receiving a vector of class probabilities ($f_v(X_i) = y_i = (p_{i1}, p_{i2}, ..., p_{ik})$) in response. The adversary repeats this to build an attacker dataset D_A and trains a substitute model f_c to clone f_v's functionality.

the QNN model accepts and how it presents its output. Furthermore, the adversary remains uninformed about the quantum device(s) on which the circuit was trained or on which device the circuit executes during the inference phase—when they query the model. Using only this limited input-output data, the adversary attempts to construct an adversarial dataset D_A. Their aim is to train a cloned model, leveraging this dataset, which can mimic the cloud-hosted QNN's functionality.

2.2 Attack Strategy

In Fig. 1, we illustrate the complete attack procedure which resembles classical model stealing techniques. Given input, X_i, consisting of features $(x_{i1}, x_{i2}, ..., x_{id})$, where d represents the number of features per sample which the victim QNN takes as input, an adversary sends queries to a cloud-based, trained QNN. The victim QNN, represented as $f_v(X_i)$, executes the consequent circuit on a quantum device. The quantum device returns measured expectation values for the qubits as output. Subsequently, these values are fed into a classical linear layer. Notably, the number of neurons in this layer equals the total class count. This layer then produces the softmax probability vector, y, for each class. The combination of X_i and y_i allows the adversary to generate an adversarial dataset that mirrors the functionality of the victim model, f_v. Once in possession of this dataset, the adversary can train a cloned QNN model, f_c. The intent is to make this clone, f_c, operate in a manner closely resembling the victim QNN, symbolized as $f_v \sim f_c$.

2.3 Defense Strategies

Model stealing attacks are challenging to defend against [2]. Defense strategies often involve perturbing the output of the victim model to hinder the attacker's ability to train a clone model using noisy datasets from the target. In quantum computing, NISQ devices already exhibit inherent noises. These noises can be leveraged to disrupt output values and protect cloud-hosted QNNs. We utilize the variability of NISQ devices to counter model stealing attacks through two main defense techniques: a) Hardware Variation-Induced Perturbation (HVIP), and b) Hardware and Architecture Variation-Induced Perturbation (HAVIP) (Fig. 2).

a) HVIP: Rather than consistently executing the quantum circuit on the same device for each attacker query, the victim can dynamically alternate between various quantum devices that differ in terms of coupling maps, basis gates, readout errors, and gate errors naturally perturbing the output vectors. These inherent variations can obscure the measurement output, thereby making it harder for the attacker to train a clone model that effectively mirrors the functionality of the cloud-based QNN. In classical model stealing techniques, this isn't feasible because the hardware used for training, such as GPUs are ideal i.e., they lack inherent noise. Consequently, executing the model on different GPUs during the inference stage would yield identical outputs.

b) HAVIP: To further enhance the robustness of cloud-based QNNs against model stealing attacks, the victim can secretly train multiple QNN models on different devices and host them on the cloud. This concealed setup keeps attackers unaware of the exact number and types of QNN models available. Each incoming query is processed using a randomly selected QNN executed on specific quantum hardware, producing varying outputs. This strategy, known as HAVIP, provides significant obfuscation by utilizing diverse QNN architectures that generate different probability distribution vectors when executed on various devices. These differences are amplified post-transpilation, as compiled circuits can drastically differ in gate counts, depths, and errors, like swap gate numbers, qubit quality, and total gate error. For example, using two distinct QNN architectures each trained on a separate device can yield a noisy dataset for the attacker, complicating the training of a cloned model.

3 EVALUATION

3.1 Setup

Training: For evaluation, we used PQC-1,6,17,19 from [7] to build our victim and clone model (PQC-x represents circuit-x in [7]), initialized with random weights. For training QNN when only the top label is available we used the NLLLoss() and used KLDivergence()

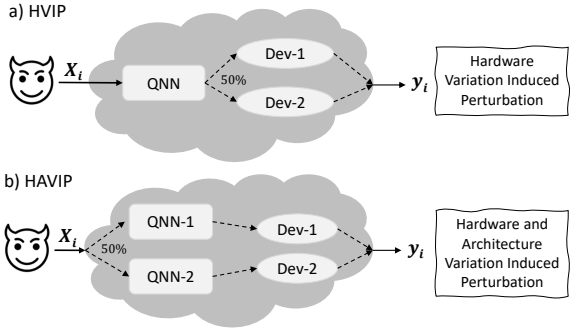

Figure 2: Proposed defense techniques which leads to obfuscated output values.

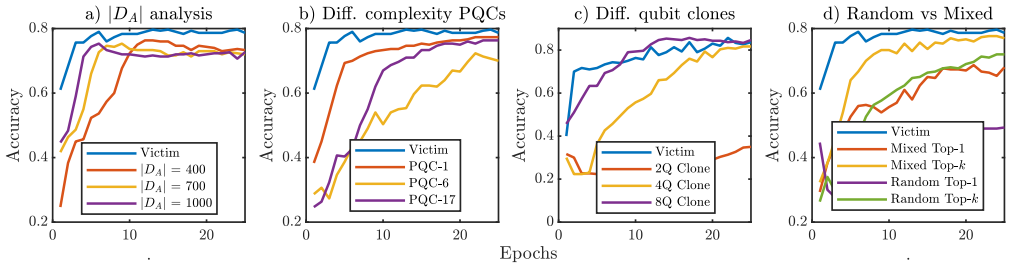

Figure 3: Plots comparing test accuracies for clone models; a) trained using different sized datasets ($|D_A|$) , b) with different sized PQCs i.e. different circuit depth and gate count but same qubit count, c) with different width QNNs i.e. different qubit clone models and d) when trained using mixed i.e., merging NPD datasets vs random dataset.

when the full probability vectors is available. The hyperparameters used for training both victim and clone model are; Epochs: 25, learning_rate (η): 0.01, batch_size = 32 and optimizer: Adam.

Device: We conducted our experiments using Pennylane's "default.mixed" device, introducing custom noises such as readout error, amplitude damping, phase flip, depolarizing error, and BitFlip error, with error probabilities of 0.01 to 0.1, to mimic real quantum hardware. We also added custom basis gates similar to those in IBMQ devices to ensure our simulations reflected actual quantum circuit performance post-compilation. Given the infeasibility of traditional gradient calculation methods like backpropagation on quantum devices, we employed the Simultaneous Perturbation Stochastic Approximation (SPSA) method to generate noisy gradients that represent realistic quantum hardware performance.

Dataset: We conduct all experiments using a reduced feature set of MNIST, Fashion, Kuzushiji and Letters datasets with latent dimension $d = 8$ (from original 28×28 image) generated using a convolutional autoencoder [1]. Thus, for each dataset, we create a smaller 4-class dataset from these reduced feature sets i.e., MNIST-4 (class 0, 1, 2, 3), Fashion-4 (class 6, 7, 8, 9), Kuzushiji-4 (class 3, 5, 6, 9) and Letters-4 (class 1, 2, 3, 4) with each having 1000 samples (700 for training and 300 for testing). For creating the unlabeled dataset which attacker uses for querying the victim QNN, we use a mix of non-problem domain (NPD) datasets. For example, if the victim QNN is trained on MNIST-4, we use an unlabeled dataset formed by mixing features of Fashion-4, Kuzushiji-4 and Letters-4 and similarly for the other cases. Since, the original model is trained on 700 samples, we used datasets of size exactly 700 to query and train our victim model.

Table 1: The final test accuracy and loss for both the victim and clone QNN after 25 epochs of noisy training on various datasets. It is evident that training with probability vectors (top-k, k: num_classes) yields significantly better performance than using only the predicted label (Top-1).

	Victim		Clone (Top-1)		Clone (Top-k)	
Datasets	Acc.	Loss	Acc.	Loss	Acc.	Loss
MNIST-4	0.896	0.371	0.726	0.691	0.880	0.547
Fashion-4	0.856	0.475	0.776	0.625	0.823	0.590
Kuzushiji-4	0.796	0.709	0.680	0.782	0.776	0.749

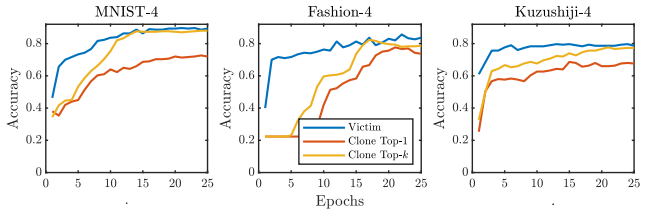

Figure 4: Comparison of test accuracy between the victim and the clone model trained with Top-1 and Top-k labels across various datasets. The clone model, when trained using the Top-k vector, closely mirrors the performance of the victim due to the richer information per training sample.

3.2 Results and Analysis of Attack

Top-1 vs. Top-k: We evaluated the efficacy of model stealing attacks on hybrid QNNs under two conditions. Cloud-based QNN returns: 1) only the highest probability (Top-1) or 2) the complete probability vector for all classes (Top-k). The results from Table 1 and Fig. 4 distinctly show that the cloned model, when trained with Top-k labels, aligns more closely with the performance of the victim model than when trained using solely the Top-1 label. This can be attributed to the richer information offered by the full probability vectors. Such comprehensive data equips the attacker with a more descriptive dataset, allowing the cloned model to mirror the cloud-based QNN's more accurately.

$|D_A|$ Analysis: We also experimented with different sized D_A in-order to see how big of an impact does the size of D_A (represented as $|D_A|$) have on the performance of clone model. This is particularly important since size of $|D_A|$ is directly proportional to the total cost incurred by the attacker to query and create D_A. If smaller sized dataset can be used, that would allow for more efficient model stealing. From Fig. 3a) we can clearly see that eventhough larger dataset allows faster learning, the final test accuracy is similar for all cases. Thus, it reflects that the attacker is better off using smaller sized dataset compared to one used for training since larger dataset does not necessarily improve the performance of the clone model.

Different PQCs: Next, we compared the clone performance of models with different architectures. PQC-1 here is the least complex model with no entangling gates, followed by PQC-17 and PQC-6 being the most complex circuit in terms of circuit depth, gate count and entanglement [7]. The results from Fig. 3b) show that the simpler circuit perform better than the more complex PQC-6 which indicates that simpler quantum circuits are better for model stealing. One possible explanation for the same might be that simpler model has lesser depth and gate counts thus the total error accumulated, especially after compilation is lower than the complex models.

Architecture Width: Given that the attacker lacks knowledge about the architecture of the cloud-based QNN, they are also unaware about the qubit count of the target model. Thus, we ran experiments with clone models of varying qubit counts (width). As depicted in Fig. 3c), the 8Q clone model outperformed others, while the 2Q model lagged behind. Hence, an intuitive strategy for an attacker would be to use a high-qubit clone model for training.

Mixed vs. Random: In a real-world context, a user of a cloud-based model would typically understand the objective of the QNN—whether it's classification, or another tasks. This understanding would enable them to craft an augmented mixed dataset from related datasets, potentially with analogous images or features. However, if an attacker lacks access to such datasets, their alternative would be to generate a random feature set for querying the victim model. Yet, as observed in Fig. 3d), datasets based on random features might not be a practically viable option since it considerably underperforms compared to the mixed dataset.

3.3 Results and Analysis of Defense Strategies

Defense Configuration: For the HVIP evaluation, we trained a cloud-based QNN primarily on one device and then partially on another to make it resilient to both devices' noise characteristics. Upon query, the victim randomly selects a device to execute the model and provide the output probabilities. In the HAVIP scenario, we trained two different QNN architectures (PQC-1 & PQC-19) on separate devices. During inference, the victim randomly operates one of these QNNs on the trained device. This process leads the attacker to inadvertently generate a noisy adversarial dataset D_A, used to train clone models. Notably, each device has unique characteristics such as noise levels, error rates, and basis gates.

Perturbation Analysis: In Table 2, we compare the obfuscation levels of HVIP and HAVIP against a baseline scenario without defense, where a single cloud-based QNN consistently runs on the same hardware. We evaluate Top-1 label mismatches and Top-k predictions' total variation distance (TVD) between the defensive strategies and the standard attack scenario. Our results show that variations in hardware and architecture alone can introduce perturbations up to 15.71%. Increasing the number of devices or the count of trained QNNs could enhance obfuscation further.

Performance Evaluation: In Fig. 5, we evaluate the efficacy of cloned models across various datasets with and without HVIP and HAVIP defenses. Results show that performance of clones with defenses is comparable to, or slightly lower than, those without defenses. For example, on the Fashion-4 dataset, the clone model's test accuracy dropped by about ~ 13% with HAVIP. However, performance is similar across other datasets and techniques. A possible reason is that these models are trained in noisy environments with

Table 2: Perturbation added by HVIP and HAVIP for both Top-1 and Top-k scenarios on various datasets.

	HVIP		HAVIP	
Datasets	Top-1	Top-k	Top-1	Top-k
MNIST-4	10.29%	6.1%	**11.2%**	**8.3%**
Fashion-4	8.14%	5.6%	**15.71%**	**10.2%**
Kuzushiji-4	8.43%	4.7%	**14.29%**	**6.4%**

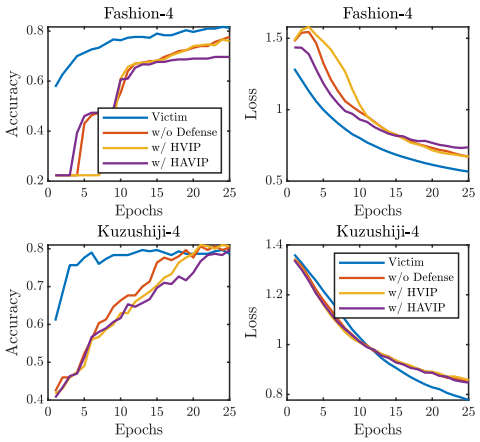

Figure 5: Test accuracy and loss comparison of cloud-based vs. clone models across various datasets, trained using Top-k labels, with/without proposed defense strategies.

gradients computed via SPSA, which might enhance resilience to minor perturbations. This aligns with strategies used in classical DNNs, where training with added noise improves resistance to adversarial attacks by enabling the model to learn in noisy conditions. Consequently, this renders hardware or architecture-based perturbation methods of limited effectiveness in practice.

4 CONCLUSION

In this study, we explored the effectiveness of model stealing attacks and defenses on QNNs. We observed that such attacks could generate clone QNNs that achieve, on average, 0.95× the test accuracy of the original victim model. To safequard QNNs from such attacks, we evaluate the effectiveness of two defensive methods that capitalize on the intrinsic noise and variability of NISQ devices, aiming to obfuscate the output probability vectors. Our work revealed a key insight: QNNs trained in a noisy environment possess an inherent resilience to defenses or attacks based on perturbations/obfuscations. *This study highlights that model theft could be a significant security issue for emerging QML platforms.*

ACKNOWLEDGMENTS

The work is supported in parts by the National Science Foundation (NSF) (CNS-1722557, CCF-1718474, OIA-2040667, DGE-1723687, and DGE-1821766) and Intel's gift.

REFERENCES

[1] Mahabubul Alam et al. 2021. Quantum-classical hybrid machine learning for image classification (ICCAD special session paper). In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–7.
[2] Sanjay Kariyappa et al. 2020. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 770–778.
[3] Sanjay Kariyappa et al. 2021. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13814–13823.
[4] Daryna Oliynyk et al. 2023. I know what you trained last summer: A survey on stealing machine learning models and defences. *Comput. Surveys* (2023).
[5] Saiwa. [n. d.]. *Machine Learning as a Service (MLaaS) | Everything you need to know about that.* https://saiwa.ai/blog/mlaas-1/
[6] Tegjyot Singh Sethi and Mehmed Kantardzic. 2018. Data driven exploratory attacks on black box classifiers in adversarial domains. *Neurocomputing* (2018).
[7] Sukin Sim et al. 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies* (2019).