Q-Detection: A Quantum-Classical Hybrid Poisoning Attack Detection Method

Haoqi He , **Xiaokai Lin** , **Jiancai Chen** and **Yan Xiao**[*]

School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University

{hehq23, linxk5, chenjc26}@mail2.sysu.edu.cn, xiaoy367@mail.sysu.edu.cn

Abstract

Data poisoning attacks pose significant threats to machine learning models by introducing malicious data into the training process, thereby degrading model performance or manipulating predictions. Detecting and sifting out poisoned data is an important method to prevent data poisoning attacks. Limited by classical computation frameworks, upcoming larger-scale and more complex datasets may pose difficulties for detection. We introduce the unique speedup of quantum computing for the first time in the task of detecting data poisoning. We present Q-Detection, a quantum-classical hybrid defense method for detecting poisoning attacks. Q-Detection also introduces the Q-WAN, which is optimized using quantum computing devices. Experimental results using multiple quantum simulation libraries show that Q-Detection effectively defends against label manipulation and backdoor attacks. The metrics demonstrate that Q-Detection consistently outperforms the baseline methods and is comparable to the state-of-the-art. Theoretical analysis shows that Q-Detection is expected to achieve more than a 20% speedup using quantum computing power.

1 Introduction

In modern image classification tasks, training classification models typically requires a large amount of data, and this data often comes from external sources such as web scraping, data marketplaces, or crowdsourcing platforms. While this method of data acquisition brings rich data resources, it also introduces potential security risks: malicious data providers can manipulate data to degrade model performance or interfere with its prediction behavior as [Wang *et al.*, 2022] said.

Data poisoning attacks are a common form of data contamination. These attacks alter the training data (features, labels, or both) to degrade model performance or exert control over it. Numerous studies as [He *et al.*, 2020; Schwarzschild *et al.*, 2021] have found that even simple noise or malicious data points can significantly impact deep learning models. When

it comes to critical domains such as autonomous driving or medical diagnosis, such attacks can severely mislead image classification models, leading to serious safety hazards.

Data poisoning attacks mainly include the following forms:

- **Label-only attacks**: Only modify the labels without changing the features, such as Target Label-Flipping Attacks [Jha *et al.*, 2023].

- **Feature-only attacks**: Only manipulate the features without changing the labels, such as Narcissus Backdoor Attacks [Zeng *et al.*, 2023b].

- **Label-Feature attacks**: Modify both labels and features simultaneously, such as BadNets attacks [Gu *et al.*, 2019].

To counter these attacks, researchers have proposed a series of defense methods, including data augmentation, adversarial training, anomaly detection, etc [Kumar *et al.*, 2020; Wang *et al.*, 2019; Zeng *et al.*, 2022].

However, these methods usually rely on **clean benchmark datasets** for training or validation. Data poisoning detection methods on large datasets may encounter large time complexity. In recent years, quantum technology has shown potential for acceleration in complex optimization tasks [He and Xiao, 2025]. By using quantum methods in the detection process, it is possible to obtain dual benefits in terms of efficiency and performance against poisoning. Therefore, how to extract a clean subset by quantum method to avoid interference from poisoned samples to model performance has become a critical issue.

In this work, we explore the intersection of quantum computing and data poisoning detection, designing a quantum-classical hybrid poisoning attack detection method to filter clean training datasets for image classification tasks. This method ensures that models trained on the clean subset achieve leading accuracy and are expected to benefit from quantum computational speedups in the future.

We propose **Q-Detection**, a quantum-classical hybrid poisoning attack detection method. Q-Detection is based on the fact that the distribution between poisoned and clean data shifts, leading to higher losses during cross-validation. It transforms the poisoning detection problem into a bilevel optimization problem, maximizing the loss of the model trained on one subset when evaluated on another subset. Q-Detection

---

ensures model training occurs in a clean environment by eliminating poisoned samples, resulting in deep learning models with higher accuracy.

Q-Detection establishes a **Quantum Weight-Assigning Network (Q-WAN)** that can be trained using quantum devices. For the first time, it transforms the selection problem into a QUBO optimization problem for the Q-WAN, enabling training with different quantum devices (such as quantum annealers, coherent optical quantum computers, superconducting gate model quantum computers). This provides a new direction for integrating quantum computing into deep learning in the future.

When filtering clean data, the Q-WAN will train alongside a domain model. Utilizing the domain model's prediction errors during training on unknown datasets allows the network to understand sample features. By inputting different data to generate different prediction errors, the Q-WAN's parameters are dynamically adjusted, enabling different weights to be assigned to different data samples. These weights are then used to filter and identify clean samples, suppressing poisoned samples.

Our contributions are as follows:

- Q-Detection enhances the integration of quantum computing and security fields. To our knowledge, it is the first quantum-classical hybrid method to transform the selection problem into a bilevel QUBO optimization problem, introducing quantum computational power. It is the first quantum method to complete data poisoning detection and clean dataset selection tasks.

- Q-Detection proposes a QUBO-based bilevel formulation method that serves as a unifying backbone, enabling training of the Q-WAN using different quantum computing technologies.

- Q-Detection achieves SOTA and consistently outperforms baseline methods across three different poisoning attacks and various poisoning ratios, demonstrating its advanced capabilities. Furthermore, the selected subset used to train new models achieves accuracy surpassing baseline methods on both global and specific label data.

- Q-Detection is enhanced by the progress of the times: our experiments reveal that the hidden layer size of the Q-WAN, linearly correlated with the number of qubits, determines Q-Detection's selection capability. Thus, it can be anticipated that with breakthroughs in quantum computing technology and the gradual increase in qubits, Q-Detection will have a broader prospect in the future.

2 Preliminaries

This section briefly reviews representative instances in Label-only, Feature-only, and Label-Feature attacks (such as Random Label-Flipping, Narcissus Backdoor Attacks, BadNets). It then provides a concise introduction to the fundamentals of quantum optimization, including QUBO problems, the Ising model, the optimization principles of different quantum devices, and the implementation of neural networks trained using quantum computing.

2.1 Attack Principles

Targeted Label-Flipping

Targeted Label-Flipping attacks strategically alter the labels of training data, causing the model to learn incorrect label distributions during training and thereby affecting classification accuracy during testing.

BadNets

BadNets attacks involve inserting triggered samples into the training set. These samples have specific triggers (e.g., pixel patterns in images) added and are relabeled to a target class.

Narcissus Backdoor Attacks

Narcissus Backdoor Attacks insert samples into the training data that use "salient regions" as triggers. This method combines salient features of input samples, making the backdoor trigger naturally integrated with the original sample features, thereby enhancing the stealthiness of the attack.

2.2 QUBO Problems and Ising Model

In the tutorial of [Glover *et al.*, 2019] QUBO is a general mathematical modeling method that transforms complex optimization problems into a quadratic polynomial form:

$$\min \mathbf{x}^\top Q \mathbf{x} + \sum_i h_i x_i, \tag{1}$$

where $\mathbf{x}$ is a binary decision variable vector, Q is a symmetric weight matrix, and h_i is the external penalty term for individual variables.

We consider an Ising system whose energy function can be expressed as:

$$E = \sum_{i>j} J_{ij}\sigma_i\sigma_j + \sum_i h_i\sigma_i, \tag{2}$$

where σ_i represents the spin state of the i-th qubit, taking values ± 1. J_{ij} is the coupling strength between spins, and h_i is the external magnetic field. Generally, the minimum value of E is sought, and the corresponding variable arrangement is determined in the process of minimization, that is, the value of σ_i.

In the Ising model, the spin variables $\sigma_i \in \{-1, +1\}$ and the binary variables $x_i \in \{0, 1\}$ in the QUBO problem can be mapped through the following relationships:

$$x_i = \frac{1 + \sigma_i}{2}, \quad \sigma_i = 2x_i - 1. \tag{3}$$

2.3 Optimization of QUBO Problems Using Different Quantum Devices

This subsection introduces three types of quantum devices capable of optimizing QUBO problems: Quantum Annealer, Coherent Quantum Photon Computer, and Gate Model Superconducting Quantum Computer.

Quantum Annealer (QA)

QAs by D-Wave [Headquarters, 2020] are devices suitable for directly inputting coefficient matrices to solve QUBO problems. They simulate the quantum annealing process found in nature to search for global optimal solutions. Works by [Rajak *et al.*, 2023; Morita and Nishimori, 2008] show that thanks to quantum tunneling effects, QAs can more easily find optimal solutions.

Coherent Quantum Photon Computer (CQPC)

CQPCs, especially Coherent Ising Machine (CIM) by QBoson Inc. [Xu *et al.*, 2023; Chi, 2024], utilize the interference and entanglement characteristics of optical components to explore the solution space. Benefiting from optical design, CQPCS can perform searches from low to high energy states, thereby quickly finding the lowest energy state, which corresponds to the optimal solution [Zhong *et al.*, 2020; Langford *et al.*, 2011; Inagaki *et al.*, 2017].

Gate Model Superconducting Quantum Computer

Studies of [Choi and Kim, 2019; Zhou *et al.*, 2023; Qian *et al.*, 2024] have shown that Gate Model Superconducting Quantum Computers are capable of running the Quantum Approximate Optimization Algorithm (QAOA). By transforming the QUBO problem into a quantum circuit and utilizing quantum gate operations to search for possible solutions, QAOA can be effectively executed on gate model quantum computers. Qiskit by IBM Inc. [Cross, 2018] is one of QAOA simulators.

2.4 Quantum Methods for Training Neural Network

The famous term quantum machine learning, as shown in the study of [Du *et al.*, 2025], uses quantum circuit. Recently, several studies ([Scellier *et al.*, 2024; Gao and Qiu, 2022; Li *et al.*, 2019]) have shown that neural networks can be trained using energy-based methods. Among them, quantum computing devices can usually obtain a quantum solution by minimizing energy, and use the quantum solution to optimize the weight of the neural network to achieve the purpose of training the neural network. [Laydevant *et al.*, 2024] has been using the ising machine and energy-based method to train neural networks.

3 The Design of Q-Detection

This section introduces the design of Q-Detection, presenting the complete workflow and optimization methods in Fig.1. It also explains how the Q-WAN incorporates quantum computing through a bilevel QUBO optimization framework.

3.1 Bilevel Optimization Process

To formalize the above idea, we define the following optimization problem:

Suppose we have a poisoned dataset $D = \{(x_i, y_i)\}_{i=1}^n$, where x_i represents the input sample and y_i is the corresponding label. We divide D into two subsets: B and $D \setminus B$, where B is the clean dataset we aim to extract.
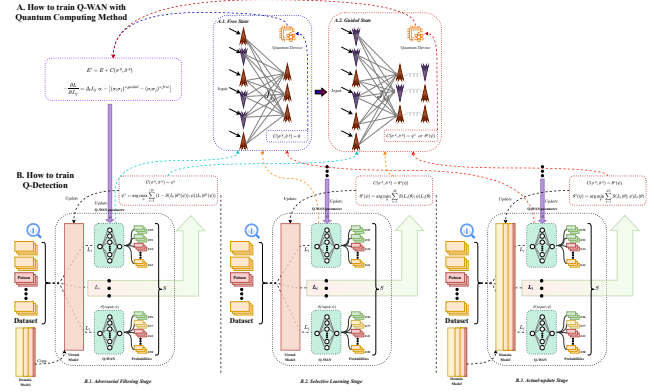

Figure 1: Q-Detection uses a one-hidden-layer network which is trained by quantum methods to filter poisoned data by assigning probabilities to samples. It maximizes the error for poisoned data in the Adversarial Filtering Stage and minimizes the error for clean data in the Selective Learning Stage and Actual-update Stage.

Since binary classification cannot be directly integrated into the network training process, we relax the binary variable $v(k, i)$ indicating whether sample z_i belongs to training subset B to a continuous variable $v(k, i) \in [0, 1]$, representing the probability that sample z_i is assigned as clean data. Here, k represents the training Stage.

The goal of Q-Detection is to train a Q-WAN $S(\cdot; \psi)$ with parameters ψ, which takes the model loss $L_i(\theta)$ as input and outputs the weight for each sample. In this paper, the loss is the cross-entropy loss. This process can be formalized as the following bilevel optimization problem:

$$\psi^* = \arg\max_{\psi} \sum_{i=1}^n \left(1 - S(L_i(\theta^*(\psi)); \psi)\right) L_i(\theta^*(\psi)),$$

$$\text{subject to} \quad \theta^*(\psi) = \arg\min_{\theta} \sum_{i=1}^n S(L_i(\theta); \psi) L_i(\theta). \quad (4)$$

where: θ represents the domain or its virtual model parameters and ψ represents the parameters of Q-WAN. $L_i(\theta)$ denotes the loss of model θ on sample z_i. $S(L_i(\theta); \psi)$ is the weight-assigning network outputting the weight for sample z_i.

Considering the difficulty of solving the constraint problem, we relax it as a two-step update process, including **Virtual-update** and **Actual-update** Stages. This is the process indicated by part B in Fig.1. The virtual-update Stage has two sub-stages including **Adversarial Filtering**, **Selective Learning** Stages. The purpose of the virtual-update Stage is to give Q-WAN a training buffer so that it can assess whether incoming data is poisoned, preventing any direct harm to the domain model, and thus allowing Q-Detection to train properly. Q-WAN will participate in every stage, learning how to assign weights to data by our QUBO model and the training rules of Q-WAN.

Among them, the domain model replicates the virtual model, and the virtual model and Q-WAN jointly perform training in the Adversarial Filtering and Selective Learning

Stage. The Actual-update Stage is jointly trained by the domain model and Q-WAN. This is because if Q-WAN is not well-trained, executing Eq.4 on the domain model will cause severe poisoning to the domain model, affecting the performance of Q-Dection.

Adversarial Filtering Stage In this Stage, we **maximize** the weighted error sum of data with a high probability of being poisoned (low-weight samples), penalizing the virtual model from learning from poisoned data. The optimization objective is:

$$\arg\max_{\psi} \sum_{i=1}^{n} \left(1 - S(L_i(\theta^*(\psi)); \psi)\right) L_i(\theta^*(\psi)) \quad (5)$$

Selective Learning Stage In this Stage, we **minimize** the weighted error sum of data with a high probability of being clean (high-weight samples), encouraging the virtual model to prioritize learning from clean data. The optimization objective is:

$$\arg\min_{\theta} \sum_{i=1}^{n} S(L_i(\theta); \psi) L_i(\theta) \quad (6)$$

Actual-update Stage In this Stage, the same optimization of Selective Learning Stage is taken, however encouraging the domain model to prioritize learning from clean data. The optimization objective is similar to Eq. 6.

By alternating the optimization of ψ and θ, we gradually improve the model's adaptability to clean data while reducing dependence on poisoned data.

The above process is similar to Meta-Sift, but we introduce a Q-WAN, transforming the training of the weight-assigning network into a quantum optimization problem.

3.2 QUBO Model of the Q-WAN

Q-WAN training encodes the network as a QUBO, samples a "free" low-energy state and a "guided" state nudged toward target outputs on a quantum annealer, computes weight-wise correlation differences between the two states, and applies these differences as gradient updates in an iterative quantum-classical loop.

To transform the training process into a QUBO optimization problem, we redefine the training method of the weight-assigning network. By representing the training of the weight-assigning network as a QUBO problem, we can utilize quantum computing devices to solve it, achieving a quantum-classical hybrid poisoning detection method.

To incorporate **guided excitation** into the energy function of the Ising system, we modify the energy function to:

$$E' = E + C(\sigma^y, \hat{\sigma}^y), \quad (7)$$

where:

- E is as defined in Equation2.
- y is the index set of output neurons.
- $\hat{\sigma}_i$ is the target spin state (guided state).
- $C(\sigma^y, \hat{\sigma}^y)$ represents the guided excitation term.

When the system is in the ground state (free evolution), $C(\sigma^y, \hat{\sigma}^y) = 0$; when guided excitation is introduced, $C(\sigma^y, \hat{\sigma}^y) = \psi^*$ (Adversarial Filtering Stage) or $C(\sigma^y, \hat{\sigma}^y) = \theta^*(\psi)$ (Selective Learning Stage and Actual-update Stage).

By introducing guided excitation into the energy function, we can minimize the spin energy during training, thereby optimizing the weight-assigning network. The introduction of guided excitation causes differences between the free state and guided state of the system, which can be used to compute gradients.

3.3 Training Rules and Gradient Calculation

To optimize the above objective, we design an innovative learning method. Using an equilibrium propagation-like approach, we derive the gradient direction from the discrete solution space by comparing the differences of the Q-WAN in the Free Stage and Guided Stage, and use it to update the parameters of the Q-WAN.

Here, we use the Mean Square Error (MSE) as the loss function, representing the deviation between the activation functions of the output neurons and their target states in the Q-WAN. We define two different measures L_1 and L_2 to evaluate the discrepancy between the activation states of output neurons and the target states, mathematically expressed as:

$$L_1(\rho(\mathbf{y}), \rho(\hat{\mathbf{y}})) = \arg\min_{\theta^*(\psi)} \|\rho(\mathbf{y}) - \rho(\hat{\mathbf{y}})\|^2, \quad (8)$$

$$L_2(\rho(\mathbf{y}), \rho(\hat{\mathbf{y}})) = \arg\min_{-\psi^*} \|\rho(\mathbf{y}) - \rho(\hat{\mathbf{y}})\|^2, \quad (9)$$

where:$\mathbf{y}$ represents the set of output neurons. $\rho(\mathbf{y})$ denotes the activation function outputs of neurons.$\hat{\mathbf{y}}$ is the target output state.

Measuring and Recording Spins We measure and record the spins at the end of the free and guided stages. The following learning rule is used to measure and record the changes in the states of output neurons and compute the gradient of the loss function with respect to the coupling strength J_{ij}:

$$-\frac{\partial L}{\partial J_{ij}} = \Delta J_{ij} \propto - \left[(\sigma_i \sigma_j)^{\text{guided}} - (\sigma_i \sigma_j)^{\text{free}}\right], \quad (10)$$

where: $(\sigma_i \sigma_j)^{\text{guided}}$ and $(\sigma_i \sigma_j)^{\text{free}}$ represent the spin pairs in the guided state and free state, respectively.

We use these updates to adjust the parameters of the Q-WAN, allowing the network to gradually approach the optimal state. This is the process indicated by part A in Fig.1. The different states have solved by quantum devices and recorded the spins. The Q-WAN uses Eq.10 to update.

4 Experiment

In this section, we design a series of experiments to verify the effectiveness of Q-Detection in defending against poisoning attacks and filtering clean data. The experiments are based on the GTSRB traffic sign image dataset, employing various common poisoning attack methods such as Targeted Label-Flipping attacks, BadNets attacks, and Narcissus backdoor

attacks [Jha *et al.*, 2023; Zeng *et al.*, 2023b; Gu *et al.*, 2019] to evaluate the performance of Q-Detection.

The aim of the experiments is to, after poisoning the dataset, use defense methods like Q-Detection to filter out 4,000 clean data samples for training classification models (ResNet-18). Finally, the filtered datasets are used to train the same deep learning models, and the model performance is evaluated to test the quality of the filtered datasets.

4.1 Experimental Setup

Metrics: **Corruption Ratio (CR)** is a common metric used to measure the proportion of poisoned data in a dataset. The CR is calculated as:

$$CR = \frac{N_{\text{Poi}}}{|D_{\text{Sub}}|} \times 100\%, \tag{11}$$

where N_{Poi} is the number of poisoned data points, and $|D_{\text{Sub}}|$ denotes the size of the selected subset D_{Sub} from D using an automated method.

Normalized Corruption Ratio (NCR) is used as the evaluation metric to measure the ability to filter out poisoned data. NCR matrices normalize away the imbalance when detection performance is measured, allowing apples-to-apples comparisons across different attack methods. The NCR is calculated as:

$$NCR = \frac{CR}{CR_{\text{rand}}} \times 100\%, \tag{12}$$

where CR_{rand} is the Corruption Ratio of a randomly selected subset, representing the capability of random filtering and serving as a random baseline.

A NCR value above 100% indicates that the selection performance is worse than the naive random baseline. In contrast, a NCR value below 100% implies better selection than random. The best possible NCR is 0%, indicating that the selected data is totally clean.

NCR value is used instead of attack success rate (ASR) to evaluate the quality of detection methods, because in most experiments, the SOTA method can achieve 0 poisoning detection. This means that the model will not be directly attacked.

General Settings All experiments were conducted on a computer with an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz with 64 GB RAM and an NVIDIA GeForce RTX 3090 24GB GPU. We utilized the Kaiwu SDK, D-Wave, and Qiskit Python API to solve the QUBO problem. Due to resource constraints, we mainly conducted experiments in a simulated environment. The PyTorch framework and cuda was used for training classic models.

We considered the popular benchmark dataset **GTSRB** (German Traffic Sign Recognition Benchmark) in [Stallkamp *et al.*, 2011]. GTSRB is widely used for traffic sign recognition, containing images of 43 categories of traffic signs, with 39,209 training images and 12,630 test images.

We used **ResNet-18** as the base model for training on the GTSRB dataset. ResNet-18 is a lightweight deep convolutional neural network with good performance and generalization ability.

Attack Type	Poisoning Ratio (%)	DCM	Loss Scan	AutoEncoder Outlier	Meta-Sift	Q-Det. CQPC 500 qubits	Q-Det. QA 500 qubits	Q-Det. QA 5000 qubits
Targeted Label-Flipping	3	15.24%	0.00%	76.58%	**0.00%**	0.00%	0.00%	**0.00%**
	5	18.06%	0.00%	108.91%	**0.00%**	0.00%	0.00%	**0.00%**
	10	38.17%	0.00%	106.55%	**0.00%**	0.00%	0.00%	**0.00%**
	20	62.27%	5.61%	115.65%	**0.00%**	0.00%	0.00%	**0.00%**
	30	87.99%	6.59%	118.25%	**0.00%**	0.00%	0.00%	**0.00%**
Narcissus Backdoor	3	0.00%	94.39%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	5	0.00%	132.57%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	10	0.00%	235.59%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	20	0.00%	197.89%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	30	0.00%	188.47%	0.00%	**0.00%**	180.00%	267.00%	**0.00%**
BadNets Backdoor	1	0.00%	0.00%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	3	30.49%	0.00%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	5	18.05%	0.00%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	8	16.24%	0.00%	0.00%	**0.00%**	0.00%	0.00%	**0.00%**
	10	16.95%	0.00%	0.00%	**0.00%**	110.14%	330.00%	**0.00%**
	30	57.21%	121.07%	0.00%	**0.00%**	231.24%	365.57%	**0.00%**

Table 1: Q-Det. is the abbreviation of Q-Detection in this table. We compare several defence methods against three types of poisoning attacks while varying the available qubits (500 and 5000). Each Q-WAN uses the same number of hidden-layer neurons as qubits and is trained with simulated quantum computing sdk. Because CQPC simulated SDK currently supports no more than 600 qubits, only the 500-qubit setting is reported for that solver. At 500 qubits, both Q-Det. CQPC and Q-Det. QA fall behind the SOTA method on backdoor attacks with higher poisoning ratios. In contrast, Q-Det. with 5000 hidden layer neurons trained on 5000 qubits performs significantly better under higher levels of attack and achieves SOTA at 0% NCR, illustrating how Q-Detection gains from advances in quantum-computing hardware.

4.2 How Clean the Selected Sub-dataset is

For the GTSRB dataset, we applied Targeted Label-Flipping attacks, BadNets attacks, and Narcissus backdoor attacks to poison the data. Then, we used various filtering methods to remove poisoned data, comparing the NCR metrics of Q-Detection with other methods such as DCM and Meta-Sift.

We filtered out 4,000 data samples as the clean dataset.

In Q-Detection, we set 20, 500, and 5000 neurons as the hidden layer of the Q-WAN, corresponding to the number of qubits in the simulation. We used local devices and the Kaiwu SDK (600 qubits limitation), D-Wave SDK (5000+ qubits limitation), and Qiskit (30 qubits limitation) for solving QUBO. Packages that exceed the qubits limit were not used.

After training, we obtained the trained Q-WAN and used it to filter clean data. Comparing with other filtering methods, we obtained the results shown in Table1. It is worth noting that in the experiment with 20 neurons, the three training methods all failed. Qiskit's QAOA method could only train 20 neurons and obtained a 929% NCR, which is due to the small size of the hidden layer and the insufficient representation ability of the neural network. Additionally, [Chu *et al.*, 2023; Li *et al.*, 2025] showing that quantum circuits are inherently vulnerable to circuit poisoning attacks, we propose using Quantum Annealer or CQPC with more qubits.

The tables show the NCR metrics of different defense methods under three representative poisoning attacks. The SOTA method Meta-Sift achieves the best NCR metrics under all attack types, demonstrating significant advantages in filtering clean data.

Q-Detection maintains an advantage over other baseline methods and achieves SOTA with 5000 qubits. Based on the success of simulated quantum computing, it shows potential for faster solutions on real quantum hardware in the future.

Other baseline methods have various drawbacks. The

(a) OVERALL ACCURACY UNDER TARGET LABEL-FLIPPING ATTACK

POISON RATE	3	5	10	20	30
DCM	67.49%	62.02%	61.95%	45.30%	57.73%
LossScan	83.17%	83.80%	84.34%	76.97%	79.58%
AutoEncoderOutlier	51.77%	54.82%	47.09%	45.84%	44.82%
Meta-Sift	90.95%	86.67%	91.09%	90.75%	90.58%
Q-Detection QA	88.99%	91.68%	89.06%	91.52%	88.95%
Q-Detection CQPC	93.24%	91.74%	90.43%	93.53%	91.51%

(b) ACCURACY OF LAB = 38 UNDER TARGET LABEL-FLIPPING ATTACK

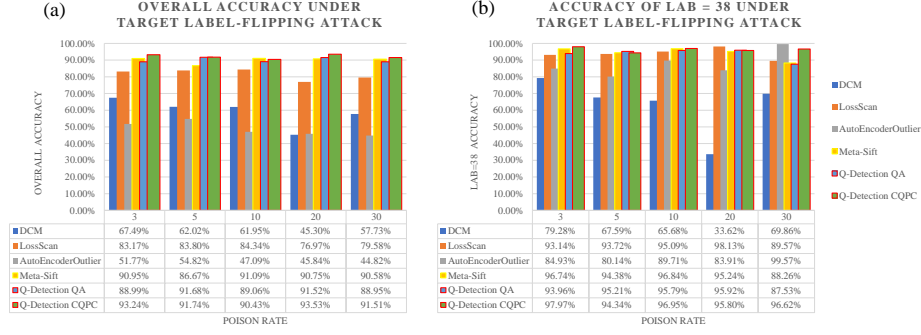| POISON RATE	3	5	10	20	30
DCM	79.28%	67.59%	65.68%	33.62%	69.86%
LossScan	93.14%	93.72%	95.09%	98.13%	89.57%
AutoEncoderOutlier	84.93%	80.14%	89.71%	83.91%	99.57%
Meta-Sift	96.74%	94.38%	96.84%	95.24%	88.26%
Q-Detection QA	93.96%	95.21%	95.79%	95.92%	87.53%
Q-Detection CQPC	97.97%	94.34%	96.95%	95.80%	96.62%

Figure 2: Under the Target Label flipping attack, we consider the poisoning ratio from 3% to 30%, and Q-WAN has 500 hidden layer neurons, which consumes 500 qubits, so both QA and CQPC methods can train Q-WAN. 4000 clean data are extracted using the trained Q-WAN to form the clean data subset, and then a new ResNet-18 is trained using the clean data subset to test its accuracy of overall and Target label equal to 38.

DCM method is only effective against Narcissus Backdoor attacks and performs poorly against other attacks. The LossScan method is extremely ineffective in filtering Narcissus Backdoor attacks. The Autoencoder Outlier method performs poorly under BadNets Backdoor attacks and takes about 30 minutes to train, which is longer than Q-Detection and Meta-Sift.

4.3 Train a New Model on the Selected Sub-Dataset

After filtering, the selected sub-dataset is used for training a brand new deep learning model and testing its accuracy. Higher model accuracy demonstrates high-quality of sub-dataset.

The experimental results in Fig.2 show that Q-Detection keeps ahead of Meta-Sift in overall accuracy, achieving the SOTA effect. In the accuracy test of label is 38, Q-Detection trained with CQPC still keeps the lead over Meta-Sift. Q-Detection is superior to SOTA method most of the time, for example, when poison rate is 20% and 30%, Compared with the model trained on the dataset screened by Meta-Sift, the overall accuracy of Q-Detection CQPC is improved by about **3%** and **1%** in Fig.2 part (a).

4.4 A Theoretical Quantitative Estimate of the Superiority of Q-Detection Time

Since Q-Detection requires multiple calls to quantum computing devices, and current quantum computing devices are not well-integrated with GPUs, we perform a theoretical estimation of Q-Detection under the Targeted Label-Flipping Attack to speculate on the potential acceleration that quantum computing can bring to the field. We run Q-Detection and Meta-Sift multiple times, take their average values, and use the average simulated computing time as a reference.

We conducted tests on QBoson Inc's 550-qubit CQPC, obtaining an average time difference between simulated quantum computing and real machines of 10^3. This means that 1 second of simulated quantum computing corresponds to only 1 millisecond on a real machine. Similarly, on D-Wave's Quantum Annealer, the real machine computation time is significantly less than that of simulated quantum computing using classical computers [Willsch et al., 2022; Honjo et al., 2021; Xu et al., 2023].

Note: The quality of solutions differs; simulated quantum computing typically provides near-optimal solutions but does not guarantee the optimal solution, whereas real machines often find the optimal solution.

We set the batch size to 179, requiring each Weight-Assigning Network to train for 220 epochs. Using the simulated CQPC, the overall training time averages 19 minutes. In contrast, Meta-Sift's average training duration is 1 minute. This discrepancy is primarily due to the slow nature of simulated quantum computing.

Each batch of data training necessitates four calls to quantum computing devices. For the Kaiwu SDK simulating CQPC, this requires an average of 4.5 seconds per call, totaling 880 calls to quantum computing devices and spending 16.5 minutes on simulated quantum computing. Therefore, we estimate the total training time with integrated CIM computing power to be $0.0165 + 2.5 = 2.52$ minutes, which is still slower than Meta-Sift accelerated purely by CUDA.

Notably, using simulated QA can achieve an average training speed of 1 minute and 10 seconds, which is already very close to the CUDA-accelerated Meta-Sift training time. This suggests promising direct acceleration effects on real Quantum Annealer hardware.

Reducing the batch size lowers memory requirements, necessitating more training epochs and increasing the number of calls to quantum computing devices.

When we set the batch size to approximately one-quarter of the original size, i.e., 44, each Weight-Assigning Network needs to train for 892 epochs, resulting in an overall training time of 69 minutes. In contrast, Meta-Sift's average training duration is 3 minutes. Each batch of data training still consumes 4.5 seconds of computation time, requiring a total of 3,568 calls to quantum computing devices and spending 66.9 minutes on simulated quantum computing. Thus, we estimate the total training time with integrated CIM computing power to be approximately $2.1 + 0.0669 = 2.17$ minutes, achieving over a 20% acceleration compared to Meta-Sift purely accelerated by CUDA.

We believe that under a smaller batch size, the acceleration

advantage over pure CUDA is more obvious, which is similar to choosing a relatively small batch size on a larger data set. Therefore, Q-Detection will show more advantages when the data volume is increasing.

5 Related Work

To address the impact of data poisoning on image classification models, researchers have proposed various defense methods, especially in the tasks of clean data selection and anomaly detection.

Some typical methods attempt to distinguish poisoned samples from clean samples by analyzing sample features, loss patterns, or data distribution characteristics. The following are several representative research methods:

1. **DCM (Distance to Class Means)**

   Method Overview: The DCM method of [Fredrikson *et al.*, 2015] is a clustering-based approach. It extracts the output vectors of samples from the last hidden layer of a proxy model as feature representations. By calculating the distance between each sample and the mean of its class, clean samples are filtered out. It is a simple and effective method, which we consider as a baseline method.

   Specifically, the mean sample (class center) of each category is first calculated, and then the distance from each sample to the class center is computed. Samples with smaller distances are regarded as clean samples.

   Advantages: Simple and efficient, particularly suitable for datasets with regular feature distributions.

   Disadvantages: May perform poorly in scenarios with imbalanced class distributions or severely poisoned data.

2. **LossScan (Poisoning Scan Based on Feature Loss Sensitivity)**

   Method Overview: LossScan of [Li *et al.*, 2021] utilizes the loss characteristics of samples in the early stages of training to determine whether they are poisoned. Studies have shown that poisoned samples often exhibit abnormal losses during early training. By recording training losses and filtering out samples with abnormally high losses, poisoned samples can be effectively removed.

   Advantages: Simple to implement, no need for complex modeling, can efficiently utilize information generated during the training process.

   Disadvantages: Relies on the training process, increasing training time.

3. **Autoencoder-based Outlier Detection**

   Method Overview: The autoencoder-based outlier detection method by [Zhou and Paffenroth, 2017] constructs an unsupervised model to learn low-dimensional representations of the data and uses reconstruction error to identify anomalous samples. Anomalous samples typically have higher reconstruction errors and can thus be filtered out as potential poisoned samples.

   Advantages: Does not require label information, suitable for unsupervised scenarios, and has strong adaptability.

   Disadvantages: In some scenarios, autoencoders may overfit to anomalous samples, affecting detection performance.

4. **Meta-Sift**

   Method Overview: Meta-Sift by [Zeng *et al.*, 2023a] innovatively proposes a bilevel optimization framework to extract clean subsets from poisoned datasets. The inner optimization objective is to minimize the training loss of the selected subset, while the outer optimization objective is to maximize the prediction loss on the unselected data. Through this mechanism, poisoned samples can be effectively identified, enhancing the robustness of the classification model.

   Meta-Sift is a novel and efficient method, which we consider as the state-of-the-art (SOTA) method.

   Advantages: Does not rely on clean benchmark data, can provide efficient defense against various poisoning attacks.

   Disadvantages: The computational complexity of bilevel optimization is high, suitable for scenarios with abundant computational resources.

The above methods provide different perspectives and technical solutions for filtering clean datasets. Existing methods rely on classical computing frameworks and may be limited by computational resources and efficiency when handling high-dimensional data and complex scenarios. Many works([Cerezo *et al.*, 2022; Ruefenacht *et al.*, 2022; He, 2024]) in the field of quantum computing have demonstrated the potential of quantum computing to accelerate the solution of various problems.

However, to our knowledge, the application of quantum computing in the defense or detection of data poisoning attacks is very rare [West *et al.*, 2023a; West *et al.*, 2023b].

6 Conclusion

Through simulated experiments and theoretical estimations, we have verified the acceleration that integrating quantum computing can bring to the field of data poisoning detection. Furthermore, we reiterate that the Q-Detection method benefits from the development of quantum computing technology, as evidenced by the enhanced representation capability of the Q-WAN with more qubits and improved compatibility of the quantum-classical hybrid architecture. In the future, real machine experiments are expected to bring actual acceleration capabilities.

References

[Cerezo *et al.*, 2022] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.

[Chi, 2024] Ming Chi. Application study of simulated annealing solver and cim simulator based on qubo model using kaiwu sdk. In *2024 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, pages 902–907. IEEE, 2024.

[Choi and Kim, 2019] Jaeho Choi and Joongheon Kim. A tutorial on quantum approximate optimization algorithm (qaoa): Fundamentals and applications. In *2019 international conference on information and communication technology convergence (ICTC)*, pages 138–142. IEEE, 2019.

[Chu *et al.*, 2023] Cheng Chu, Lei Jiang, Martin Swany, and Fan Chen. Qtrojan: A circuit backdoor against quantum neural networks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[Cross, 2018] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March meeting abstracts*, volume 2018, pages L58–003, 2018.

[Du *et al.*, 2025] Yuxuan Du, Xinbiao Wang, Naixu Guo, Zhan Yu, Yang Qian, Kaining Zhang, Min-Hsiu Hsieh, Patrick Rebentrost, and Dacheng Tao. Quantum machine learning: A hands-on tutorial for machine learning practitioners and researchers. *arXiv preprint arXiv:2502.01146*, 2025.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[Gao and Qiu, 2022] Xiangyu Gao and Meikang Qiu. Energy-based learning for preventing backdoor attack. In *International Conference on Knowledge Science, Engineering and Management*, pages 706–721. Springer, 2022.

[Glover *et al.*, 2019] Fred Glover, Gary Kochenberger, and Yu Du. A tutorial on formulating and using qubo models. 2019.

[Gu *et al.*, 2019] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[He and Xiao, 2025] Haoqi He and Yan Xiao. Probabilistic quantum svm training on ising machine. *arXiv preprint arXiv:2503.16363*, 2025.

[He *et al.*, 2020] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. Towards security threats of deep learning systems: A survey. *IEEE Transactions on Software Engineering*, 48(5):1743–1770, 2020.

[He, 2024] Haoqi He. Quantum annealing and gnn for solving tsp with qubo. In *International Conference on Algorithmic Aspects in Information and Management*, pages 134–145. Springer, 2024.

[Headquarters, 2020] Corporate Headquarters. Technical description of the d-wave quantum processing unit, 2020.

[Honjo *et al.*, 2021] Toshimori Honjo, Tomohiro Sonobe, Kensuke Inaba, Takahiro Inagaki, Takuya Ikuta, Yasuhiro Yamada, Takushi Kazama, Koji Enbutsu, Takeshi Umeki, Ryoichi Kasahara, et al. 100,000-spin coherent ising machine. *Science advances*, 7(40):eabh0952, 2021.

[Inagaki *et al.*, 2017] T. Inagaki, K. Inaba, and R. Kawamura. Coherent ising machines—optical neural networks operating at the quantum limit. *npj Quantum Information*, 3:23, 2017.

[Jha *et al.*, 2023] Rishi Jha, Jonathan Hayase, and Sewoong Oh. Label poisoning is all you need. *Advances in Neural Information Processing Systems*, 36:71029–71052, 2023.

[Kumar *et al.*, 2020] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE security and privacy workshops (SPW)*, pages 69–75. IEEE, 2020.

[Langford *et al.*, 2011] Nathan K Langford, Sven Ramelow, Robert Prevedel, William J Munro, Gerard J Milburn, and Anton Zeilinger. Efficient quantum computing using coherent photon conversion. *Nature*, 478(7369):360–363, 2011.

[Laydevant *et al.*, 2024] Jérémie Laydevant, Danijela Marković, and Julie Grollier. Training an ising machine with equilibrium propagation. *Nature Communications*, 15(1):3671, 2024.

[Li *et al.*, 2019] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6243–6251, 2019.

[Li *et al.*, 2021] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

[Li *et al.*, 2025] Y. Li, R. Du, and Y. Zhang. Adversarial data poisoning attacks on quantum machine learning. *arXiv preprint*, 2025.

[Morita and Nishimori, 2008] Satoshi Morita and Hidetoshi Nishimori. Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12), 2008.

[Qian *et al.*, 2024] Yang Qian, Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. Mg-net: Learn to customize qaoa with circuit depth awareness. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Rajak *et al.*, 2023] Atanu Rajak, Sei Suzuki, Amit Dutta, and Bikas K Chakrabarti. Quantum annealing: An

overview. *Philosophical Transactions of the Royal Society A*, 381(2241):20210417, 2023.

[Ruefenacht *et al.*, 2022] Martin Ruefenacht, Bruno G Taketani, PASI Lähteenmäki, VILLE Bergholm, DIETER Kranzlmüller, LAURA Schulz, and MARTIN Schulz. Bringing quantum acceleration to supercomputers. *IQM/LRZ Technical Report, https://www. quantu m. lrz. de/fileadmin/QIC/Downloads/IQM HPC-QC-Integration-White paper. pdf*, 2022.

[Scellier *et al.*, 2024] Benjamin Scellier, Maxence Ernoult, Jack Kendall, and Suhas Kumar. Energy-based learning algorithms for analog computing: a comparative study. *Advances in Neural Information Processing Systems*, 36, 2024.

[Schwarzschild *et al.*, 2021] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021.

[Stallkamp *et al.*, 2011] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[Wang *et al.*, 2019] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pages 707–723. IEEE, 2019.

[Wang *et al.*, 2022] Zhibo Wang, Jingjing Ma, Xue Wang, Jiahui Hu, Zhan Qin, and Kui Ren. Threats to training: A survey of poisoning attacks and defenses on machine learning systems. *ACM Computing Surveys*, 55(7):1–36, 2022.

[West *et al.*, 2023a] Maxwell T West, Sarah M Erfani, Christopher Leckie, Martin Sevior, Lloyd CL Hollenberg, and Muhammad Usman. Benchmarking adversarially robust quantum machine learning at scale. *Physical Review Research*, 5(2):023186, 2023.

[West *et al.*, 2023b] Maxwell T West, Shu-Lok Tsang, Jia S Low, Charles D Hill, Christopher Leckie, Lloyd CL Hollenberg, Sarah M Erfani, and Muhammad Usman. Towards quantum enhanced adversarial robustness in machine learning. *Nature Machine Intelligence*, 5(6):581–589, 2023.

[Willsch *et al.*, 2022] Dennis Willsch, Madita Willsch, Carlos D Gonzalez Calaza, Fengping Jin, Hans De Raedt, Marika Svensson, and Kristel Michielsen. Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems. *Quantum Information Processing*, 21(4):141, 2022.

[Xu *et al.*, 2023] Hui-Zhang Xu, Jun-Hua Chen, Xing-Chen Zhang, Te-Er Lu, Tian-Ze Gao, Kai Wen, and Yin Ma. High-speed train timetable optimization based on space–time network model and quantum simulator. *Quantum Information Processing*, 22(11):418, 2023.

[Zeng *et al.*, 2022] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022.

[Zeng *et al.*, 2023a] Yi Zeng, Minzhou Pan, Himanshu Jahagirdar, Ming Jin, Lingjuan Lyu, and Ruoxi Jia. {Meta-Sift}: How to sift out a clean subset in the presence of data poisoning? In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1667–1684, 2023.

[Zeng *et al.*, 2023b] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 771–785, 2023.

[Zhong *et al.*, 2020] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[Zhou and Paffenroth, 2017] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[Zhou *et al.*, 2023] Zeqiao Zhou, Yuxuan Du, Xinmei Tian, and Dacheng Tao. Qaoa-in-qaoa: solving large-scale max-cut problems on small quantum machines. *Physical Review Applied*, 19(2):024027, 2023.