Data Poisoning Quantization Backdoor Attack

Tran Huynh[1], Anh Tran[1], Khoa D. Doan[2], and Tung Pham[1]

[1] VinAI Research, Vietnam
{v.tranhn2, v.anhtt152, v.tungph4}@vinai.io
[2] College of Engineering and Computer Science, VinUniversity, Vietnam
khoa.dd@vinuni.edu.vn

Abstract. Deep learning (DL) models are often large and require a lot of computing power. Hence, model quantization is frequently used to reduce their size and complexity, making them more suitable for deployment on edge devices or achieving real-time performance. It has been previously shown that standard quantization frameworks can be exploited to activate the backdoor in a DL model. This means that an attacker could create a hijacked model that appears normal and free from backdoors (even when examined by state-of-the-art defenses), but when it is quantized, the backdoor is activated, and the attacker can control the model's output. Existing backdoor attack methods on quantization models require full access to the victim model, which might not hold in practice. In this work, we focus on designing a novel quantization backdoor based on data poisoning, which requires zero knowledge of the target model. The key component is a trigger pattern generator, which is trained together with a surrogate model in an alternating manner. The attack's effectiveness is tested on multiple benchmark datasets, including CIFAR10, CelebA, and ImageNet10, as well as state-of-the-art backdoor defenses.

Keywords: Backdoor attacks · Data poisoning · Quantization backdoor

1 Introduction

Deep learning models have achieved remarkable performance on many tasks, especially in vision and language. However, success is often achieved by using extremely large models, which are unsuitable for resource-constrained devices such as mobile phones. To make DL models deployable for everyone regardless of their limited computing resources, a practical solution called *model quantization* has been proposed [8]. The main idea of quantization is to reduce the numerical precision of the weights and activations in the deep learning model in order to save memory and also decrease the computation cost in arithmetic operations. These compression techniques [14, 20, 22, 45] have been integrated into many popular ML frameworks, including PyTorch, TensorFlow, TensorRT, and Core ML, under the expectation that the quantized models still maintain relatively good performance. However, the trade-off for computational cost is not always negligible as anticipated. Recent works have found that the quantized model might behave differently from the original model, which leads to new security

issues that could be exploited by malicious attackers. Hence, it is of utmost significance and interest to understand the danger of model quantization from potential attacks.

In this paper, we focus on backdoor attacks, an increasingly important security threat to AI systems. Traditional backdoor attacks aim to fool the victims into acquiring backdoor models, either via model pretraining or data poisoning. These models are disguised as normal models by performing genuinely on clean data. However, when some specific patterns, called backdoor triggers, appear in the input, the models will return incorrect outcomes as defined by the attackers. The attackers trick the victims into employing such models in their systems, thereby acquiring system controls for illegal benefits or destructive purposes. This security threat is becoming increasingly critical since it is hard to build an advanced AI system without using any third-party resources nowadays. Hence, backdoor has drawn many research interests in recent years, on both attack [5, 12, 15, 27, 31, 32, 50] and defense [4, 21, 25, 26, 39, 42, 47–49].

As of the time of writing this paper, there have been over 120 proposed defense techniques specifically designed for countering backdoors in image and video classification tasks [1]. Being aware of the backdoor threat, a knowledgeable user could employ some of these defenses to verify a full AI model before its deployment, thus significantly reducing the risk. However, as previously mentioned, the adversary can trick the user to deploy a special type of model that behaves normally at full precision but acquires backdoors after quantization. That tricky behavior can bypass the security measures of even knowledgeable users to cause atrocious damage. Unfortunately, such an attack mechanism, called Quantization backdoor attack, has rarely been investigated in the literature [19, 33, 38].

Given the subtle behavior difference between full-precision and quantized models, it is challenging to design effective quantization-based backdoor attacks. All existing approaches require complete control of model training so that they can employ either attack-based training losses [19, 33, 38] or model optimization [19]. Hence, they can only work on the easy attack scenario in which the attackers act as model providers, and the victims use the provided full-precision models as is for quantization. That scenario has limited applications; it will be more trustworthy for the end-users if they get involved in model training. We will reveal that it is still a false sense of security since it is possible to construct a quantization-based attack without any interference to the training process.

Our proposed attack in this paper is through **data poisoning**. This attack is more practical and pervasive since the attackers just have to provide some training data while the victims have full control over the model development. More particularly, the attacker only needs to modify a small part of the training data, either in only the content (clean-label) or in both the content and label (dirty-label), so that any AI model trained on such data will be benign on full precision but has the backdoor behavior after being quantized. Notably, clean-label attackers only inject the backdoor patterns to data but keeps their label unchanged, making it stealthier but more challenging to design. We aim to develop quantization backdoor attacks for both configurations. Our focus is

on quantization backdoor attacks in image classification, a common target in backdoor research, with a methodology easily extendable to other domains.

Designing a quantization backdoor attack without control of the model training process is challenging. To address this, we propose two key components: using backdoor triggers from a trigger generator network instead of random patterns and training the trigger generator with a surrogate of the target model in an alternated manner. We make no assumptions about the victim model's architecture, allowing flexibility. Our method is evaluated on CIFAR-10, ImageNet-10, and CelebA, demonstrating effectiveness and robustness. These results highlight the need for practitioners to develop defenses against such potential attacks.

The outline of this paper is as follows. In Sec. 2, we review the literature on backdoor attacks/defenses and quantization. In Sec. 3, we formulate the problem and present our attack scheme in detail. Sec. 4 shows our method's experiment results and ablation studies. The main paper ends with conclusions and future work discussions in Sec. 5. Additional results can be found in the Appendix.

2 Background

2.1 Threat model

In backdoor attacks, the attacker could provide the victim with a poisoned dataset (dataset-poisoning) or a poisoned network (model-poisoning). In this work, we study the scenario where the attacker pretends to be an open-source or commercial data provider that supplies a victim with a dataset to build a model for image classification. Before releasing the data set to the victim, the attacker poisoned the dataset by injecting a pre-defined trigger into a small subset of the data. The trigger pattern could be in any form, such as noise, image patch, blended content, or image warping. The classification model obtained from training on these poisoned data will produce correct label of normal input even if the input is embedded with trigger. Thus, the victim does not recognize any malicious behavior from the obtained model and then quantizes and deploys it in his or her system. However, once the model is quantized, the backdoor behavior of the model will be activated, allowing the attacker to gain illegal benefits.

Unlike previous quantization-based attacks, our method imposes fewer assumptions on attackers' required knowledge. Attackers don't need control over the victim's training process or prior knowledge of the target network architecture and training procedure. While attackers may anticipate the quantization framework the victim would use, e.g., models developed with PyTorch are typically compressed by PyTorch Mobile, they do not need to know the exact quantization method or the calibration datasets used for model quantization.

2.2 Previous backdoor attacks

Data poisoning backdoor attacks could be categorized into two groups: dirty-label and clean-label. The most popular backdoor attacks are dirty-label attacks,

where a trigger is injected into an image, whose label is changed to the target class. The simplest dirty-label backdoor attack is BadNets [15] when the trigger is a fixed pattern embedded in certain locations of an image, which could be easily recognized by a human. After BadNets, many methods have been proposed to design stealthy triggers that are difficult to detect. [5] blended a fixed trigger into images; [36] varied randomly the locations and patterns of triggers; [31] designed data-dependent triggers to run away from the previous fixed-trigger assumption. There also have been a lot of efforts to make the trigger imperceptible. [32] used image warping; [12] optimized the trigger in the input space; [11, 50] created the trigger from the latent space; [16, 43] did it in the frequency domain.

Dirty-label attacks use incorrect labels for poisoned data, which could be detected as mislabeled data and consequently will be ignored or relabeled. [40] is the first to discuss this issue and then investigate if the attack is still successful while maintaining label consistency. It turned out that the classifier is likely to recognize the backdoor trigger as a feature of the targeted class, thus it will assign any datum having that trigger to that class. Other attacking methods have been recommended to create natural-looking triggers; for example, using texture in image [35], using image reflection [28], matching gradient [37] of losses between normal and poisoned data, or making the feature more invisible by minimizing distance between poisoned and clean data in feature space [35].

2.3 Backdoor defense methods

To defend against backdoor attacks, the victim could apply protection methods in all or a stage in the process of building a model, ranging from dataset scanning (***data defense***) and trained model examination (***model defense***) to test-time monitoring when the model is already deployed (***test-time defense***). Defensive methods in each stage are briefly summarized below.

Data defense. The victim aims to identify then purify or remove potentially poisoned data samples to prevent the backdoor formation from the source. Based on the fact that the trigger acts as a repeated feature in poisoned data, [39] detects backdoor samples through projecting them on the singular vector of data matrix in the feature space; [4] relied on clustering in the feature space; [48] identified poisoned data based on the high frequency artifacts.

Model defense. At this stage, defenders look for the abnormal model's behaviors when dealing with clean and poisoned data; it could raise suspicion then discard, deactivate, or make adjustments to the model. Notable defense methods include Fine-pruning [25], which prunes inactive neurons even without direct backdoor evidence. ABS [26] reverse-engineers backdoor triggers from neurons and tests them on clean data. Neural Cleanse [42] computes optimal patterns for each class, detecting suspicious small patterns indicative of backdoor triggers. Other approaches include examining the gradient of classification score changes [46], using optimized input images with a meta-classifier [21], employing distillation to highlight backdoor regions in the attention map (NAD) [24], and viewing the backdoor trigger as a min-max problem (I-BAU) [47].

Test-time defense. Before performing inference on the data with the model, the defender can try to filter out malicious samples by looking at potential trigger regions, e.g., Neo [41] and [10] (GradCAM), or exploiting some properties of neural networks on the perturbed data [13], STRIP.

2.4 Quantization methods

Quantization-based compression techniques work by reducing the number of bits needed to represent parameters of a deep learning model, thus decreasing model size and computational cost. Early works perform quantization by clustering the weights of a model and representing each cluster with its centroid [6, 14, 45]. Later on, [3] used 16-bit precision values to represent 32-bit floating-point model weights. This technique reduces the model size by half while maintaining a relatively similar accuracy to that of the full-precision model. It is also simpler, yet more effective than the clustering-based methods. More recently, [20] have lowered the precision to 8-bit, and the computations are mainly performed using 8-bit integer arithmetic. 8-bit quantization is considered to have the best computational performance, as an 8-bit quantized model can achieve 2-3 times speed-up during inference compared to the original full-precision model [22]. Moreover, it is supported by most popular deep learning frameworks, such as PyTorch. Therefore, in this work, we perform attacks on 8-bit quantization.

2.5 Quantization backdoor attacks

Recent works exploit the gap between full-precision and quantized models for backdoor attacks. [33] and [19] use dedicated cross-entropy loss items to define behaviors. [38] enhances attacks with knowledge distillation and data augmentation, producing distilled attacks. [19] proposes a two-step approach requiring full control of the model training process, applicable in model outsourcing scenarios. In contrast, we aim to design quantization backdoor attacks under the more challenging scenario of clean-label data poisoning, without controlling the victim's model training process. Unlike previous methods using simple patch-based triggers, we learn nearly imperceptible and input-aware trigger patterns for robust and stealthy attacks.

3 Methodology

3.1 Problem overview

In this section, we present the formulation of a black-box backdoor attack on model quantization. Let us assume that the data lie in a domain $\mathcal{X}$, which is classified into m classes denoted by $\mathcal{C}$. A clean data set is denoted by $\mathcal{S} = \big\{(x_i, y_i) : i = 1, \ldots, n\big\}$, where $x_i \in \mathcal{X}, y_i \in \mathcal{C}$ and y_i is the correct label of x_i. The user wants to adopt a classifier, says f_θ, where θ is the model's parameters, then use a quantization function $\mathcal{Q}$ to quantize the parameters $\theta_{\text{quan}} = \mathcal{Q}(\theta)$ to obtain

the quantized classifier $f_{\theta_{\text{quan}}}$ with expectation that $f_{\theta_{\text{quan}}}(x_i) = f_\theta(x_i)$ for all or most of datum x_i. Hence, if the original classifier f_θ works perfectly, the quantized classifier $f_{\theta_{\text{quan}}}$ will achieve similar performance with less computations.
Quantization Backdoor Attack: To fool the user, the desired poisoned classifier should behave genuinely at full precision, but its quantized version, while still correctly classifying clean data input, always returns the target label y_t when a backdoor trigger is present:

$$f_\theta(x) = y; \qquad f_\theta(\mathcal{B}(x)) = y; \tag{1}$$

$$f_{\theta_{\text{quan}}}(x) = y; \quad f_{\theta_{\text{quan}}}(\mathcal{B}(x)) = y_t, \tag{2}$$

where $\mathcal{B}$ is our desired backdoor injection function that alters a clean input with the backdoor trigger pattern.
Data-poisoning Quantization Backdoor Attack: The previous quantization-based attacks assumed the attacker had full control of the training process to produce f_θ. In this work, we focus on a more challenging but practical scenario in which the attacker cannot interfere with the training process but only provide a poisoned data set, denoted by $\mathcal{S}^{\text{pois}}$, for the victim to train the poisoned model f_θ. The poisoned data set $\mathcal{S}^{\text{pois}}$ is derived from the original clean dataset $\mathcal{S}$ by poisoning a small ratio $\rho \ll 1$ of the data: $(x, y) \to (\mathcal{B}(x), y_t)$, where x is benign data and y is the target class of the attack.

Data poisoning attacks are categorized as "dirty-label" and "clean-label". In "dirty-label" attacks, the attacker can poison any data sample, irrespective of the correct label y differing from the attack label y_t. In "clean-label" attacks, the attacker poisons only the target class samples ($y = y_t$), eliminating the need for data relabelling. While "dirty-label" attacks are easier and often more successful, "clean-label" attacks are stealthier and preferred in practice. Our quantization-based attacks are defined for both settings. Additionally, the attacker may have prior knowledge (white-box attack) or none (black-box attack) about the victim's training configuration. We demonstrate the effectiveness of our attacks in both white-box and black-box scenarios.

3.2 Quantization-backdoor-inducing loss

Before introducing our approach, we will briefly discuss how the previous works [19], [30] develop the quantization-based attacks. These approaches, despite their different designs, require full control of the model training process with a modified loss function, which is directly derived from the objectives described in Equations 1 and 2, to induce the desired network behaviors. This loss function is typically the weighted sum of these loss components:

$$\begin{aligned}\sum_{(x,y)\in\mathcal{S}} &\mathcal{L}_{\text{ce}}(f_\theta(x), y) + \lambda_{\text{c}}\mathcal{L}_{\text{ce}}(f_\theta(\mathcal{B}(x), y)) + \\ &\mathcal{L}_{\text{ce}}(f_{\mathcal{Q}(\theta)}(x), y) + \lambda_{\text{bd}}\mathcal{L}_{\text{ce}}(f_{\mathcal{Q}(\theta)}(\mathcal{B}(x)), y_t),\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss function, λ_{c} and λ_{bd} are weighting parameters.

Note that the objectives in Equations 1 and 2 require the full-precision and quantized models to have radically different behaviors when handling backdoor inputs. It is challenging since their corresponding outputs are supposed to be similar, given the design of quantization. To achieve the attack goal, previous methods have to force the model training to optimize the mentioned loss function; thus, full control of the training process is critically required.

However, in our setting, the attackers have no control of the training process and thus cannot directly employ these loss terms. They have to find a different way to induce the desired behavior difference between the full-precision and quantized models, which, as mentioned, is incredibly difficult. We argue that defining a proper trigger generation function $\mathcal{B}$ is the key to making that challenging task become possible.

3.3 Trigger optimization

Unlike previous quantization-based attacks that use simple, fixed patterns like white squares, we confirm in the Appendix that such random trigger patterns fail to generate victim models with distinct behaviors before and after quantization using data poisoning alone. Consequently, we propose optimizing the backdoor trigger function $\mathcal{B}$ to achieve our attack goals.

The poisoned data is often designed to be as close to the original data as possible, i.e., $\mathcal{B}(x) \approx x$. Hence, we decompose $\mathcal{B}(x)$ into two parts, including the original clean data x and the noise part $\eta b_\psi(x)$, through a simple equation $\mathcal{B}(x) = x + \eta b_\psi(x)$. The noise part is defined by its ℓ_∞ bound η and a noise function b_ψ whose value range is within $[-1, 1]$. We model the noise function b_ψ using a neural network with parameters ψ. Making this trigger function $\mathcal{B}$ learnable, we can flexibly learn to place a trigger on the input in such a way that will cause no effect on the full-precision model while inducing the quantized model's backdoor behaviors. Specifically, we wish to find ψ that minimizes the below loss function, which include f_{θ^*} the victim classifier and regularizer term imposing conditions on the trigger

$$\sum_{(x,y)\in\mathcal{S}} \Big[\mathcal{L}_{\text{ce}}(f_{\theta^*}(\mathcal{B}_\psi(x)), y) + \mathcal{L}_{\text{ce}}(f_{\theta^*_{\text{quan}}}(\mathcal{B}_\psi(x), y_t) + \mathsf{regularizer}(\eta b_\psi(x))\Big].$$

To optimize that trigger function, we need access to the victim classifier f_{θ^*}, which has not yet been trained. We can replace it with a surrogate classifier that closely mimics its behavior. Since f_{θ^*} is expected to be trained on data poisoned by $\mathcal{B}_\psi$, we let the surrogate classifier jointly train with the trigger generator in an alternated training manner. Our trigger generator optimization, therefore, includes the following steps. We first initialize a surrogate classifier f_θ and a trigger generator $\mathcal{B}_\psi$. We then alternately train these networks until convergence. The training loss for optimizing $\mathcal{B}_\psi$ is revised to be:

$$\sum_{(x,y)\in\mathcal{S}} \Big[\mathcal{L}_{\text{ce}}(f_\theta(\mathcal{B}_\psi(x)), y) + \mathcal{L}_{\text{ce}}(f_{\theta_{\text{quan}}}(\mathcal{B}_\psi(x)), y_t) + \mathsf{regularizer}(\eta b_\psi(x))\Big], \quad (4)$$

while the surrogate classifier f_θ is trained with the loss function of full-precision model to mimic the process of training the victim model on the poisoned dataset:

$$\sum_{(x,y)\in\mathcal{S}\setminus\mathcal{P}} \mathcal{L}_{\text{ce}}(f_\theta(x), y) + \sum_{(x,y)\in\mathcal{P}} \mathcal{L}_{\text{ce}}(f_\theta(\mathcal{B}_\psi(x)), y_t), \tag{5}$$

where $\mathcal{P}$ denotes the subset of $\mathcal{S}$ that is poisoned. Note that we alternatively train the losses 4 and 5, those interchange parameter θ and backdoor generator $\mathcal{B}_\psi$ instantly to secure optimal solutions for both.

After the alternated training process, the attacker obtains the optimal backdoor function $\mathcal{B}_{\psi^*}$ and uses it to generate the poisoned dataset $\mathcal{S}^{\text{pois}}$. The whole process of our dirty-label attack is described in Algorithm 1. The algorithm for generating poisoned data in the clean-label manner is provided in the Appendix.

3.4 Advanced trigger modifications

To enhance the stealthiness of the trigger, we design additional modifications for the optimization of the trigger generator. A recent study [48] argues that backdoor samples are detectable in the frequency space because backdoor triggers may introduce high-frequency noise or disrupt the natural patterns of the pixels in an image. To prevent our backdoor data from being detected by this defense, we apply extra techniques to mitigate the aforementioned issues. First, for a given noise $b_\psi(x) \in \mathbb{R}^{d\times d}$, we apply a high-frequency components filtering mask $m \in \mathbb{R}^{d\times d}$ to its type-II 2D Discrete Cosine Transform (DCT) [2] $\text{DCT}(b_\psi(x))$, where $m_{i,j} = \mathbf{1}_{1\le i,j\le rd}$, $\mathbf{1}$ is the indicator function. We take the Hadamard product of m and $\text{DCT}(b_\psi(x))$ to preserve only top-left $rd\times rd$ entries, where $r \in (0, 1)$ is a ratio parameter. The trigger noise is then reconstructed using inverse DCT (IDCT). The whole transformation can be formulated as $\mathcal{T}((b_\psi(x)) = \text{IDCT}(m \odot \text{DCT}(b_\psi(x)))$. Although the generated noise has only low-frequency components, adding it directly to an image can still decrease the correlations between neighboring pixels in the original image. To fix this, we then apply a Gaussian blur kernel k to the whole poisoned image. We have our final backdoor function $\mathcal{B}_\psi(x) = (x + \eta\mathcal{T}(b_\psi(x))) * k$.

To make the attack stealthier, we also reduce the magnitude of the trigger by minimizing the loss:

$$\mathcal{L}_{\ell_2}(b_\psi; \mathcal{S}, \eta) := \sum_{(x,y)\in\mathcal{S}} \|\eta\mathcal{T}(b_\psi(x))\|_2. \tag{6}$$

Furthermore, the generative trigger function b_ψ might create adversarial noises, which means that these perturbations can cause the quantized model's misclassification at test time without poisoning during training. We want to prevent this effect since it is not the purpose of data poisoning backdoor attacks, and adversarial defenses can mitigate these adversarial noises. Therefore, we utilize a clean classifier h_ϕ, which was pre-trained on the same task as f_θ, and constrain

Algorithm 1 Dirty-label data poisoning

Input: Training data set $\mathcal{S}$, target label y_t, injection rate p, poison magnitude η, number of training iteration N.

Stage 1: Find the optimal trigger function $\mathcal{B}_{\psi^*}$
initialize ψ and θ
for *the number of iterations* $< N$ **do**
  Randomly sample a mini-batch $\mathcal{S}_{\mathsf{mini}}$ from $\mathcal{S}$
  Randomly sample $\mathcal{P}_{\mathsf{mini}}$ from $\mathcal{S}_{\mathsf{mini}}$ with ratio p
  Update θ**:**

$$\min_{\theta} \sum_{(x_j,y_j)\in\mathcal{S}_{\mathsf{mini}}\setminus\mathcal{P}_{\mathsf{mini}}} \mathcal{L}(f_\theta(x_j), y_j) + \sum_{(x_j,y_j)\in\mathcal{P}_{\mathsf{mini}}} \mathcal{L}\big(f_\theta(\mathcal{B}_\psi(x_j)), y_t\big)$$

  Update ψ**:**

$$\min_{\psi} \sum_{(x_j,y_j)\in\mathcal{S}_{\mathsf{mini}}} \Big[\mathcal{L}\big(f_\theta(\mathcal{B}_\psi(x_j)), y_j\big) + \mathcal{L}\big(f_{\theta_{\mathrm{quan}}}(\mathcal{B}_\psi(x_j)), y_t\big) + \mathsf{regularizer}(\eta b_\psi(x_j))\Big]$$

end

Stage 2: Generate the poisoned dataset $\mathcal{S}^{\mathrm{pois}}$
Randomly sample $\mathcal{P}$ from $\mathcal{S}$ with ratio p
$\mathcal{P}^{\mathrm{pois}} \leftarrow \emptyset$
for (x, y) *in* $\mathcal{P}$ **do**

$$\mathcal{P}^{\mathrm{pois}} \leftarrow \mathcal{P}^{\mathrm{pois}} \cup \{(\mathcal{B}_\psi(x), y_t)\}$$

end
$\mathcal{S}^{\mathrm{pois}} \leftarrow (\mathcal{S} \setminus \mathcal{P}) \cup \mathcal{P}^{\mathrm{pois}}$
return $\mathcal{B}_{\psi^*}$ and $\mathcal{S}^{\mathrm{pois}}$.

$h_{\phi_{\mathrm{quan}}}$ to correctly classify all inputs, even with the trigger's presence:

$$\mathcal{L}_{\mathrm{adv}}(\mathcal{B}_\psi, h_\phi; \mathcal{S}, \eta) := \sum_{(x,y)\in\mathcal{S}} \mathcal{L}(h_{\phi_{\mathrm{quan}}}(\mathcal{B}_\psi(x)), y). \tag{7}$$

We include the loss functions above in the regularizer in Eq. (4):

$$\mathsf{regularizer}(\eta b_\psi(x)) := \lambda_{\mathrm{adv}}\mathcal{L}(h_{\phi_{\mathrm{quan}}}(\mathcal{B}_\psi(x)), y) + \lambda_{\ell_2}\|\eta\mathcal{T}(b_\psi(x))\|_2. \tag{8}$$

4 Experiments

4.1 Experimental setup

We use three popular datasets, namely CIFAR-10 [23], ImageNet-10, and CelebA [29], for our experiments. To create the ImageNet-10 dataset, we randomly select 10 classes from ImageNet-1K [9]. For CelebA, we follow the recommended configuration from [36] to choose the three most balanced attributes, namely Heavy Makeup, Mouth Slightly Open, and Smiling, and concatenate them to form eight compound classes for a multi-label classification task. To construct the surrogate classifier f, we utilize ResNet-18 [17] for all datasets. Additionally, we design the

generator function b with a U-Net [34] backbone. To acquire the quantized classifier for the trigger optimization process, we utilize static quantization-aware training [20]. For further details, please refer to the Appendix.

In each experiment, we replicate the entire data and model poisoning process, then evaluate the clean accuracy of the full-precision model, as well as the quantized model's clean and attack accuracy. In all model training, we use the SGD optimizer. The initial learning rate is 0.01, which is reduced 10 times for every 100 epochs until the model converges. We use the same target class $\mathfrak{c} = 0$ in all tests. We set λ_{ℓ_2} and λ_{adv} to 0.02 and 0.8, respectively. We evaluate our method in both dirty-label and clean-label attack scenarios. We set the poisoning rate to 5% for all datasets. To make the trigger visually imperceptible, we choose a small value of $10/255$ for the trigger magnitude η.

4.2 Attack experiments

We first conduct experiments with the standard backdoor setting, where the attacker has prior knowledge about the victim model's training and matches the surrogate model's architecture with the victim model's. For the quantization step, we adopt the 8-bit post-training static quantization. We report our method's performance in Table 1.

As shown in Table 1, across all datasets, our attack only causes a trivial decrease in benign accuracy when compared to the models trained on the clean data. As the results suggest, our victim models do not exhibit any backdoor behaviors when not quantized, as the full-precision models can still classify the poisoned samples with high accuracy. However, when the victim models are quantized, our method impressively achieves over 91% ASR on both CIFAR-10 and CelebA. On ImageNet-10, its ASR remains over 81%, demonstrating the method's effectiveness on large images.

We compare our method with previous quantization-based backdoor attacks in Table 2. Notably, our attack, operating without access to victim models' training, presents a more challenging threat model compared to previous approaches that assume full control. Nevertheless, as depicted in Table 2, our method achieves competitive ASRs with other baselines.

4.3 Transfer experiments

In practice, the attacker is unlikely to have prior information about the victim model's training and architecture. Therefore, we investigate if the backdoor behaviors induced by our attack persist when the victim model has a different architecture from the surrogate model's or when a different post-training quantization method is employed.

Table 3 shows our attack performance when we use a different architecture (i.e., VGG13) for the victim models. Additional results with MobileNet can be found in the Appendix. On CIFAR-10 and CelebA, the ASR remains over 90%. While the ASR decreases to a certain extent on ImageNet-10, it still reaches at

Table 1: Attack performance on different datasets. For the full-precision models, we report the benign accuracy (BA) in teal and the accuracy on poisoned samples in violet. For the quantized models, we report the benign accuracy (BA) in teal and the attack success rates in purple. We also report the original accuracy (OA) of the corresponding clean models as a reference.

Dataset	OA (%)	Dirty-label		Clean-label	
		Full-precision	Quantized	Full-precision	Quantized
CIFAR-10	94.38	94.20 / 90.52	94.05 / 92.32	94.31 / 90.12	94.16 / 91.52
ImageNet-10	86.52	86.44 / 77.65	86.02 / 82.37	86.48 / 78.49	86.06 / 81.27
CelebA	79.66	79.59 / 72.93	79.25 / 96.77	79.60 / 75.22	79.31 / 96.89

Table 2: Comparison with previous methods. Attacks are conducted in dirty-label manner on CIFAR-10 with ResNet18 as the victim models' backbone. For the full-precision models, we report the BA in teal and the accuracy on poisoned samples in violet. For the quantized models, we report the BA in teal and the ASR in purple.

Attack	[38]	[19]	[30]	Ours
Full-precision	93.76 / 92.05	94.09 / 90.27	93.99 / 94.32	94.20 / 90.52
Quantized	93.37 / 82.64	93.85 / 96.73	93.46 / 99.25	94.05 / 92.32

least near 75%. These results confirm that our attack is transferable to victim models with different backbones from the surrogate models.

Table 4 presents the performance of our attack's victim models when quantized with different post-training quantization methods on CIFAR-10. Despite employing static quantization-aware training for the surrogate model during trigger optimization, our attack remains effective even when the victim model is quantized using different schemes. In most cases, our attack achieves ASR exceeding 90% under both dirty-label and clean-label conditions.

Table 4: Attack performance on different quantization methods. We report the benign accuracy (BA) in teal and the attack success rates in purple of the quantized victim models.

Method	Dirty-label	Clean-label
Static	94.05 / 92.32	94.16 / 91.52
Dynamic	94.01 / 90.91	94.02 / 89.64
Weight only	93.95 / 92.55	93.89 / 90.62

4.4 Defense experiments

In this section, we provide experimental results of our clean-label attack against 5 representative defenses across different approaches: data filtering (Frequency-based defense, Spectral Signatures), backdoor model mitigation (NAD), and test-time defense (STRIP). Extra defense experiments, including Neural Cleanse, Fine-pruning, ANP, I-BAU, and GradCAM can be found in the Appendix.

Table 3: Transferability of the attack to different victim backbones. We use VGG13 as the victim architectures for the experiments. Benign accuracy (BA) is in teal. For the full-precision models, we report the accuracy on poisoned samples in violet. For the quantized models, we report the attack success rates in purple.

Dataset	Dirty-label		Clean-label	
	Full-precision	Quantized	Full-precision	Quantized
CIFAR-10	93.75 / 86.46	93.67 / 90.13	93.80 / 89.26	93.78 / 90.22
ImageNet-10	85.78 / 80.98	85.54 / 79.55	85.80 / 82.88	85.75 / 74.76
CelebA	79.02 / 71.89	78.92 / 95.23	79.11 / 75.62	78.98 / 94.48

Frequency-based defense [48] is a data defense technique that utilizes a trained detector to identify poisoned samples in frequency domain. This method is effective as many existing backdoor attacks leave high-frequency artifacts that are easily distinguishable from natural images. We address this issue by employing high-frequency removal techniques in Section 3.4. As shown in Table 5, these techniques effectively diminish the detection rate across all datasets.

Table 5: Effect of our high-frequency removal (HFR) on the detection rate (%) of frequency-based detector.

	CIFAR-10	ImageNet-10	CelebA
W/o HFR	100.00	100.00	100.00
W/ HFR	17.25	21.63	29.54

Spectral Signatures [39] utilizes *spectral signatures* to identify and eliminate backdoor inputs within the training set. Initially, a network is trained on the suspicious dataset, then used to process all samples from each label to record their latent representations. Singular value decomposition (SVD) is then applied to the covariance matrix of these latent representations, generating outlier scores. Inputs with the highest scores are recognized as poisoned samples and excluded from the training data.

Table 6: Performance against Spectral Signature (SS). Benign accuracy (BA) is in teal. For the full-precision models, we report the accuracy on poisoned samples in violet. For the quantized models, we report ASRs in purple.

Model	Before SS	After SS
Full-precision	94.31 / 90.12	94.02 / 88.87
Quantized	94.16 / 91.52	93.86 / 89.94

We assess the defense's capability to detect poisoned samples in the target class (class 0). Assuming knowledge of the target class, the defense removes samples exclusively from it using a default threshold of 15% (750 samples). The victim model is then trained with the remaining data. However, the defense only successfully detects and eliminates 56/2500 poisoned samples, main-

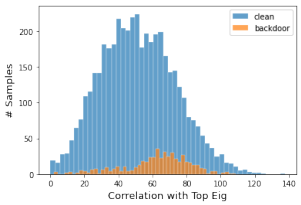

Fig. 1: Correlation of latent representations of inputs from the target class in the training dataset with the top eigenvector.

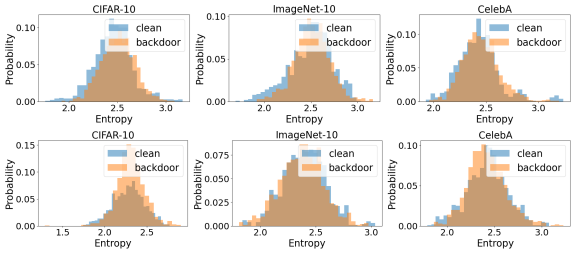

Fig. 2: Performance against STRIP of full-precision (upper row) and quantized (lower row) models.

taining high attack effectiveness (see Table 6). To understand evasion against this defense, we analyze latent representations' correlations with the top eigenvector from SVD. Figure 1 (replicated from [39]), using 4500 clean and 500 poisoned samples, reveals inseparable histograms. This suggests that removing inputs with the highest correlations is insufficient against our attack.

Neural Attention Distillation (NAD) [24] employs knowledge distillation [18] to eliminate backdoors by perturbing backdoor-related neurons. We tested our attack on CIFAR-10 using this defense. Following the original protocol, we fine-tuned the poisoned full-precision model (student) on 5% clean data for 10 epochs to create the teacher model. The NAD process was then applied, training for an additional 10 epochs. Despite a decreasing trend in ASR, indicating the defense's impact, the backdoor's effectiveness remained significant at 79.62%. Results are summarized in Table 7.

Table 7: Performance against NAD. For the full-precision models, we report the BA in teal and the accuracy on poisoned samples in violet. For the quantized models, we report the BA in teal and the ASR in purple.

Model	Before NAD	After NAD
Full-precision	94.31 / 90.12	92.96 / 89.66
Quantized	94.16 / 91.52	90.27 / 79.62

STRIP [13] is a widely-used test-time defense, which functions by overlaying various patterns onto a suspicious input and observing the model's prediction entropy for each perturbed image. Consistent predictions, characterized by low entropy values, indicate a high likelihood of the sample being poisoned. As shown in Figure 2, the backdoor models of our attack can easily bypass STRIP's detection since they have similar entropy ranges to the clean counterparts.

4.5 Ablation studies

Role of alternated training and performance w.r.t η. While the alternated training is a key component to the success of our attack, a more straightforward approach is training b using a fixed surrogate model f pre-trained on clean data. To compare the two approaches, we conduct experiments on CIFAR-10 in the clean-label settings. We report the BA and ASR of the poisoned quantized models in Figure 3. As the results indicate, our method constantly outperforms the naive one with varying trigger magnitude.

Table 8: Performance of the quantized models w.r.t poisoning rate (PR).

PR	1%	3%	5%	10%	20%
Dirty-label	94.22 / 44.89	94.12 / 83.24	94.05 / 92.32	90.55 / 95.01	85.22 / 97.06
Clean-label	94.36 / 37.48	94.29 / 77.85	94.16 / 91.52	92.86 / 93.24	87.65 / 94.93

Performance w.r.t poisoning rate. We present the influence of poisoning rate on the performance of the quantized victim models trained on CIFAR-10 in Table 8. Our method achieves around 80% ASR by poisoning only 3% of the training data (1500 samples), confirming its efficiency. As the quantity of the poisoned images grows, the ASR steadily approaches over 90%.

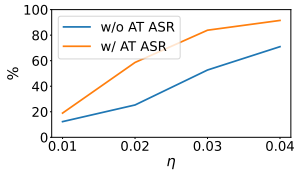

Fig. 3: Role of alternated training (AT).

Role of adversarial avoidance loss. During the trigger optimization, without the adversarial avoidance loss $\mathcal{L}_{\text{adv}}$ that we introduce in Equation 7, b tends to circumvent the intended malicious behavior by learning to generate universal targeted adversarial noises. These noises can effectively deceive the quantized victim classifier during inference, irrespective of whether data poisoning occurred during training, which contradicts the fundamental goal of data poisoning backdoor attack. Furthermore, standard adversarial defenses can effectively mitigate these adversarial noises. To illustrate this behavior, we present the ASR of our attack on CIFAR-10 with and without $\mathcal{L}_{\text{adv}}$ in Table 9. When $\lambda_{\text{adv}} = 0$, the ASR remains consistently high across different victim backbones, regardless the value of p. However, when λ_{adv} is set to 0.8, the backdoor behavior is mitigated when data poisoning is absent. This holds true even when the victim's backbone differs from the surrogate model, confirming that λ_{adv} effectively prevents b from generating adversarial noises.

Table 9: ASR (%) of our attack with and without adversarial avoidance loss.

Victim model	$\lambda_{\text{adv}} = 0$		$\lambda_{\text{adv}} = 0.8$	
	$p = 0\%$	$p = 5\%$	$p = 0\%$	$p = 5\%$
ResNet18	80.59	91.84	8.55	91.52
VGG13	72.54	91.02	10.24	90.22

5 Conclusions

In this work, we propose a novel quantization backdoor attack based on data poisoning, requiring no access to the target model. The key component of our method is the alternated training process that concurrently optimizes a trigger generator and a surrogate classifier. The attack successfully activates backdoors in quantized models trained on various datasets and bypasses state-of-the-art defenses. While previous works only study quantization backdoor attack with full model training control, our work exposes the vulnerability of quantized models to data poisoning and emphasize the need for robust defenses against such attacks.

References

1. backdoor-learning-resources (Sep 2023), `https://github.com/THUYimingLi/backdoor-learning-resources`, [Online; accessed 13. Sep. 2023]
2. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (1974)
3. Alumbaugh, T., Kulik, A., Lee, J., Duke, J., Alvarez, R., Joglekar, S., Li, J., Li, Y., Sivakumar, S., Garg, N., Chan, L., Selle, A.: Tensorflow model optimization toolkit — float16 quantization halves model size (2016)
4. Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)
5. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
6. Choi, Y., El-Khamy, M., Lee, J.: Towards the limit of network quantization. arXiv preprint arXiv:1612.01543 (2016)
7. Comon, P.: Independent component analysis (1992)
8. David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., et al.: Tensorflow lite micro: Embedded machine learning for tinyml systems. Proceedings of Machine Learning and Systems **3**, 800–811 (2021)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
10. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: Annual Computer Security Applications Conference. pp. 897–912 (2020)
11. Doan, K., Lao, Y., Li, P.: Backdoor attack with imperceptible input and latent modification. Advances in Neural Information Processing Systems **34**, 18944–18957 (2021)
12. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11966–11976 (2021)
13. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)
14. Gong, Y., Liu, L., Yang, M., Bourdev, L.: Compressing deep convolutional networks using vector quantization. arXiv preprint arXiv:1412.6115 (2014)
15. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. In: Proceedings of Machine Learning and Computer Security Workshop (2017)
16. Hammoud, H.A.A.K., Ghanem, B.: Check your other door! establishing backdoor attacks in the frequency domain. arXiv preprint arXiv:2109.05507 (2021)
17. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
18. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)
19. Hong, S., Panaitescu-Liess, M.A., Kaya, Y., Dumitras, T.: Qu-anti-zation: Exploiting quantization artifacts for achieving adversarial outcomes. Advances in Neural Information Processing Systems **34**, 9303–9316 (2021)

20. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2704–2713 (2018)
21. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: Revealing backdoor attacks in cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 301–310 (2020)
22. Krishnamoorthi, R.: Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:1806.08342 (2018)
23. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
24. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. arXiv preprint arXiv:2101.05930 (2021)
25. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 273–294. Springer (2018)
26. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1265–1282 (2019)
27. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: Proceedings of Network and Distributed System Security Symposium (2018)
28. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision. pp. 182–199. Springer (2020)
29. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
30. Ma, H., Qiu, H., Gao, Y., Zhang, Z., Abuadbba, A., Xue, M., Fu, A., Zhang, J., Al-Sarawi, S.F., Abbott, D.: Quantization backdoors to deep learning commercial frameworks. IEEE Transactions on Dependable and Secure Computing (2023)
31. Nguyen, A., Tran, A.: Input-aware dynamic backdoor attack. In: Proceedings of Advances in Neural Information Processing Systems (2020)
32. Nguyen, T.A., Tran, T.A.: WaNet - Imperceptible Warping-based Backdoor Attack. In: International Conference on Learning Representations (2021)
33. Pan, X., Zhang, M., Yan, Y., Yang, M.: Understanding the threats of trojaned quantized neural network in model supply chains. In: Annual Computer Security Applications Conference. pp. 634–645 (2021)
34. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
35. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11957–11965 (2020)
36. Salem, A., Wen, R., Backes, M., Ma, S., Zhang, Y.: Dynamic backdoor attacks against machine learning models. arXiv preprint arXiv:2003.03675 (2020)
37. Souri, H., Goldblum, M., Fowl, L., Chellappa, R., Goldstein, T.: Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. arXiv preprint arXiv:2106.08970 (2021)

38. Tian, Y., Suya, F., Xu, F., Evans, D.: Stealthy backdoors as compression artifacts. IEEE Transactions on Information Forensics and Security **17**, 1372–1387 (2022)
39. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. Advances in neural information processing systems **31** (2018)
40. Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. arXiv preprint arXiv:1912.02771 (2019)
41. Udeshi, S., Peng, S., Woo, G., Loh, L., Rawshan, L., Chattopadhyay, S.: Model agnostic defence against backdoor attacks in machine learning. IEEE Transactions on Reliability (2022)
42. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)
43. Wang, T., Yao, Y., Xu, F., An, S., Wang, T.: Backdoor attack through frequency domain. arXiv preprint arXiv:2111.10991 (2021)
44. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Advances in Neural Information Processing Systems **34**, 16913–16925 (2021)
45. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4820–4828 (2016)
46. Xu, K., Liu, S., Chen, P.Y., Zhao, P., Lin, X.: Defending against backdoor attack on deep neural networks. arXiv preprint arXiv:2002.12162 (2020)
47. Zeng, Y., Chen, S., Park, W., Mao, Z.M., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. arXiv preprint arXiv:2110.03735 (2021)
48. Zeng, Y., Park, W., Mao, Z.M., Jia, R.: Rethinking the backdoor attacks' triggers: A frequency perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16473–16481 (2021)
49. Zhao, P., Chen, P.Y., Das, P., Ramamurthy, K.N., Lin, X.: Bridging mode connectivity in loss landscapes and adversarial robustness. arXiv preprint arXiv:2005.00060 (2020)
50. Zhong, N., Qian, Z., Zhang, X.: Imperceptible backdoor attack: From input space to feature representation. arXiv preprint arXiv:2205.03190 (2022)