Targeted Bit-Width Quantization-Conditioned Backdoor

Quoc-Anh Mai*‡, Van-Thuc Le*‡, Dat-Thinh Nguyen†, Nhien-An Le-Khac† and Kim-Hung Le*§
*University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam
Email: 24520002@gm.uit.edu.vn, 24521748@gm.uit.edu.vn, hunglk@uit.edu.vn
†School of Computer Science, University College Dublin, Dublin, Ireland
Email: dat-thinh.nguyen@ucdconnect.ie, an.lekhac@ucd.ie
‡These authors contributed equally.
§Corresponding author.

***Abstract*—Quantization-aware training (QAT) is a popular approach to reducing the footprint of large image classifiers while maintaining accuracy, enabling deployments of these models on resource-constrained devices. However, QAT introduces a new attack surface, where adversaries can exploit rounding errors in quantization operators to hide backdoor behaviors that are activated only during quantization; thereby, these backdoors are called quantization-conditioned backdoors (QCBs). Despite recent advances, existing QCB methods exhibit limited stealthiness, as they activate the backdoor upon any quantization precision. In this paper, we propose a targeted bit-width QCB attack that activates only at a specific quantization precision while remaining dormant at all other precisions. Our method modifies the QAT objective to jointly preserve clean accuracy, suppress backdoor behavior at non-targeted bit-widths, while enforcing backdoor activation at the targeted bit-width by leveraging bit-width-dependent rounding discrepancies. We evaluate the proposed attack on CIFAR-10 and CIFAR-100 using MobileNetV2, ResNet34, and VGG16, and show that our method achieves high attack success at the targeted precision while keeping the attack success rate insignificant at other precisions without compromising clean accuracy.**

***Index Terms*—backdoor, quantization-aware training, quantization-conditioned backdoor**

I. Introduction

Although state-of-the-art (SOTA) image classification models have improved accuracy over time, higher accuracy often comes with substantially more parameters, which reduces inference efficiency and limits adoption in edge applications. For example, ResNet-18 uses 11.7M parameters and achieves 69.8% top-1 accuracy on ImageNet, whereas ResNet-152 requires substantially more parameters (60.2M) to reach 78.3%. Consequently, deploying SOTA image classifiers on edge devices is challenging because real-time inference typically demands substantial computational resources, dedicated GPU memory, and sufficient power to accommodate the large number of model parameters from these models [1]. To this end, model quantization [2], [3], which converts the model parameters from full-precision, i.e., 32-bit floating point (FP32), to lower-bit integer representations (e.g., INT4, INT8), is a promising approach to reduce model footprint and enable the deployment of SOTA classifiers on edge devices.

There are two common quantization approaches: post-training quantization (PTQ) and quantization-aware training (QAT). PTQ directly converts a pre-trained FP32 model into a lower-precision integer representation, without fine-tuning. In detail, PTQ typically uses a small calibration dataset and performs several forward passes to estimate quantization parameters (e.g., scale and zero-point) from activation statistics [4]. These parameters are then used to quantize activations and weights at inference time. Unfortunately, PTQ often exhibits accuracy losses at low precision (e.g., INT4), as rounding to lower-precision integers introduces larger quantization errors that can accumulate across layers, leading to significant accuracy degradation [5]. Moreover, calibration data that are not representative of the deployment distribution may yield poor activation statistics (and thus suboptimal quantization parameters), further reducing accuracy after quantization [3]. In contrast, QAT fine-tunes the model while explicitly simulating inference-time quantization in the forward pass, typically via pseudo-quantization operators [3], [6]. This enables the model to adapt its weights to quantization noise and reduces the accuracy loss compared to PTQ.

Unfortunately, recent research on adversarial attacks highlights a novel threat surface, where the QAT process is exploited to mount backdoor attacks. Under an adversarial supply chain scenario, an adversary can inject a hidden behavior into a model that preserves clean accuracy while consistently producing attacker-chosen predictions under the presence of a specific trigger pattern in the input [7]. Likewise, in an adversarial QAT scenario, the adversary injects a backdoor into a fine-tuning model; however, the quantization operator becomes an activation condition, where the backdoor remains dormant under the FP32 model and is only activated after quantization. This attack is therefore referred to as a quantization-conditioned backdoor (QCB). Consequently, QCBs become stealthier than traditional backdoor attacks as they can evade various existing FP32 validation [8], [9].

However, existing QCB methods exhibit limited stealthiness: the backdoor is activated whenever the model is quantized, regardless of the deployed precision [8], [10]. For instance, the seminal QCB attack, namely Qu-ANTI-zation [8], jointly injects the backdoor behavior across multiple quantization precisions, leading to full backdoor activation at both INT8 and INT4 quantization, and partial backdoor acti-

vation at FP32. We argue that this non-targeted bit-width QCB approach fails to remain stealthy in real-world deployments, especially under advanced persistent threat (APT) scenarios. For instance, different computing platforms prefer different precisions: Raspberry Pi AI stacks[1] and Qualcomm NPUs[2] favor INT4, whereas CUDA-enabled edge devices (e.g., DGX Spark[3] and Jetson Orin[4]) operate optimally in INT8. As a result, an APT campaign against a critical industrial system of a rival country that extensively uses Qualcomm NPUs may want their QCB to stay undetected across all other precisions, except INT4. Consequently, the stealthiness of targeted bit-width QCB attacks becomes essential for a successful APT campaign. To this end, research in targeted bit-width QCB attacks plays a critical role in understanding the dangers from this attack vector and thereby facilitates the development of robust defense techniques against this threat.

Motivated by the above limitation, we propose a targeted bit-width QCB attack[5], in which the backdoor activates only at a specific quantization precision while remaining dormant elsewhere. Our attack method is inspired by Qu-ANTI-zation, which exploits rounding errors introduced by the quantization operator when converting FP32 parameters to lower-precision integer representations. However, our method differs from Qu-ANTI-zation in its optimization objective. Specifically, we optimize a multi-precision QAT loss that enforces *contradictory trigger behaviors* across bit-widths: (i) in full precision, the model is trained to classify both clean and triggered inputs correctly (benign trigger behavior), (ii) at all *non-targeted* quantization precisions, we explicitly penalize trigger-induced misclassification so that the trigger remains ineffective after quantization, and (iii) at the *targeted* quantization precision, we enforce trigger-induced misclassification to an attacker-chosen target label while preserving clean accuracy. By jointly training these objectives under pseudo-quantization operators for multiple bit-widths, the optimization encourages weights to lie in regions where rounding discrepancies selectively activate the backdoor behavior only at the targeted quantization precision.

We evaluate the effectiveness and stealthiness of our attack on CIFAR-10 and CIFAR-100 across three model architectures (MobileNetV2, ResNet34, and VGG16) and two metrics: *clean data accuracy* (CDA) and *attack success rate* (ASR). The results demonstrate our superior stealthiness, with consistently higher ASR at the targeted quantization precision than Qu-ANTI-zation, while remaining benign at all other precisions. Notably, our targeted bit-width attack does not compromise the model's CDA, highlighting the practical impact and feasibility of targeted bit-width QCB attacks in real-world applications.

[1]https://www.raspberrypi.com/products/ai-hat-plus-2/

[2]https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/Unlocking-on-device-generative-AI-with-an-NPU-and-heterogeneous-computing.pdf

[3]https://www.nvidia.com/en-us/products/workstations/dgx-spark/

[4]https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/

[5]Code: https://github.com/uit-iec/IJCNN-2026-Targeted-Bit-Width-Quantization-Conditioned-Backdoor

II. Related Work

Quantization is a method to reduce a model's footprint by mapping FP32 parameters to low-bit integers (e.g., INT8 or INT4). While PTQ [2], [3] is straightforward, it often causes non-trivial accuracy drops due to the rounding errors [3]. On the other hand, QAT has become the dominant approach, mitigating this loss by fine-tuning the model under simulated quantization noise and thereby achieving promising results [2], [3], [6]. Unfortunately, several works show that attackers can exploit the fine-tuning pipeline to inject backdoor behaviors [11], [12]. For instance, [8], [9] introduced QCB and laid the foundation for this field, demonstrating that quantization can conceal backdoor behavior in a full-precision model but can activate it after quantization. More recently, [10], [13] analyzed the theoretical mechanisms underlying how rounding errors and gradient approximations in QAT facilitate dormant backdoors. Nonetheless, prior works exhibit limited stealthiness as the backdoor is easily revealed upon quantization to any precision. Therefore, this research aims to develop a targeted bit-width QCB attack that activates only at a specific quantization precision while remaining benign at all other precisions.

III. Problem Formulation

A. Problem Setup

We consider a classification model $f : \mathcal{X} \to \mathcal{Y}$ parameterized by W, where $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes the label set. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training dataset, where i indexes samples and N is the number of training data points, with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. We assume that the model is pretrained on $\mathcal{D}$, minimizing the standard cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$.

Subsequently, we consider a backdoor setting in which an attacker inserts a fixed trigger pattern δ onto a subset of inputs. For a clean input x, we denote the corresponding trigger-injected input by $x_t = x \oplus \delta$, where $\oplus$ denotes the trigger-insertion operator. Let $\mathcal{X}_t$ denote the set of triggered inputs, and let $y_t \in \mathcal{Y}$ denote the attacker-chosen target label; to this end, each $x_t \in \mathcal{X}_t$ is assigned the label y_t.

A pseudo-quantization operator $Q_{f_b}(\cdot)$ is used during QAT to introduce quantization noise in the fine-tuning process. Given FP32 parameters W, the pseudo-quantized parameters are defined as:

$$Q_{f_b}(W) = s \cdot \Big(\mathrm{clip}\Big(\Big\lfloor \frac{W}{s} \Big\rceil + z,\, q_{\min},\, q_{\max} \Big) - z \Big),$$

where s denotes a scaling factor, z denotes a zero point, and $q_{\min}, q_{\max}$ denote the minimum and maximum integer values for the chosen bit-width. The operation $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. Finally, we denote $\mathcal{B}$ as the set of supported quantization bit-widths. Among these, $t \in \mathcal{B}$ is the targeted bit-width and $\mathcal{C} = \mathcal{B} \setminus \{t\}$ is the set of non-targeted bit-widths.

B. Threat Model

We consider a scenario in which a user has an FP32 model pretrained on the clean dataset $\mathcal{D}$ and wants to quantize it for

edge inference. The user relies on a third-party QAT service to perform quantization-aware fine-tuning and to produce quantized models (or quantization configurations) for a set of supported precisions $\mathcal{B}$.

Attacker's capabilities: We assume an adversarial supply-chain setting in which the attacker compromises the QAT service, thereby gaining white-box access to the model and the fine-tuning pipeline. Consequently, the attacker knows the target task and can manipulate the fine-tuning objective. Additionally, the attacker has access to $\mathcal{D}$ used for fine-tuning and can construct a poisoned subset by inserting a fixed trigger δ into a fraction of inputs while assigning them the attacker-chosen target label y_t. As a result of the fine-tuning process, the attacker returns the compromised model and the associated quantization configuration to the user.

Attacker's goal: The attacker aims to realize a *targeted bit-width* backdoor: when the model is quantized to the targeted bit-width t, triggered inputs should be classified as the target label y_t (high ASR at t). Meanwhile, to maintain stealthiness, the same trigger should not activate the backdoor at any non-targeted bit-width $b \in \mathcal{C}$, and the model should preserve high accuracy on clean inputs at all precisions.

IV. Proposed Attack

A. Attack Overview

Our key idea is to constrain the backdoor to a specific bit-width quantization by explicitly optimizing for contradictory behaviors across quantization precisions during QAT. As summarized in Algorithm 1, our attack initiates from a clean pretrained model; then, we iteratively (i) sample a clean mini-batch, (ii) construct its triggered subset by injecting a fixed trigger and assigning the target label, and (iii) update the model parameters by minimizing a multi-precision objective computed under pseudo-quantization.

B. Targeted Bit-Width QAT Loss

We formulate a targeted bit-width QAT objective that jointly optimizes three requirements: (i) preserving clean accuracy, (ii) enforcing backdoor activation only at the targeted bit-width, and (iii) suppressing backdoor behavior at all non-targeted bit-widths. Concretely, the overall objective is:

$$\begin{aligned}\mathcal{L}_{\text{total}} = & \underbrace{\mathcal{L}_{ce}(f(x), y) + \lambda_2 \, \mathcal{L}_{ce}(f(x_t), y)}_{\mathcal{L}_F^c} \\ & + \underbrace{\lambda_1 \sum_{b \in \mathcal{C}} \Big(\mathcal{L}_{ce}(Q_{f_b}(x), y) + \lambda_2 \, \mathcal{L}_{ce}(Q_{f_b}(x_t), y) \Big)}_{\mathcal{L}_Q^c} \\ & + \underbrace{\lambda_1 \Big(\mathcal{L}_{ce}(Q_{f_t}(x), y) + \lambda_2 \, \mathcal{L}_{ce}(Q_{f_t}(x_t), y_t) \Big)}_{\mathcal{L}_{\text{backdoor}}}\end{aligned} \tag{1}$$

Here, λ_1 weights the overall contribution of the quantized-domain losses (i.e., $\mathcal{L}_Q^c$ and $\mathcal{L}_{\text{backdoor}}$) relative to the full-precision term $\mathcal{L}_F^c$. Meanwhile, λ_2 weights the loss on triggered samples relative to clean samples within each term, encouraging x_t to behave benignly in $\mathcal{L}_F^c$ and $\mathcal{L}_Q^c$, but mapping to y_t in $\mathcal{L}_{\text{backdoor}}$. Additionally, with an abuse of notation, we write $Q_{f_b}(x)$ for the output of model f pseudo-quantized to b bits when evaluated on input x.

Algorithm 1 Targeted Bit-Width QCB

Input: Pretrained parameters W, supported bit-widths $\mathcal{B}$, target bit-width t, steps K
Output: Compromised parameters W^*
Initialize W from the pretrained checkpoint
Define non-target set $\mathcal{C} \leftarrow \mathcal{B} \setminus \{t\}$
for $k = 1$ **to** K **do**
  Sample clean batch $(x, y) \sim \mathcal{D}$
  Construct triggered batch (x_t, y_t) with $x_t \leftarrow x + \delta$
  Compute $\mathcal{L}_F^c$ on (x, y) and (x_t, y)
  Compute $\mathcal{L}_Q^c$ over $b \in \mathcal{C}$ on (x, y) and (x_t, y)
  Compute $\mathcal{L}_{\text{backdoor}}$ at bit-width t on (x, y) and (x_t, y_t)
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_F^c + \mathcal{L}_Q^c + \mathcal{L}_{\text{backdoor}}$
  $W \leftarrow W - \eta \nabla_W \mathcal{L}_{\text{total}}$
end for
Return $W^* \leftarrow W$

$\mathcal{L}_F^c$ is the *full-precision clean loss*; it encourages the model to predict the ground-truth label y on clean inputs x and also to keep triggered inputs x_t *benign* (i.e., still classified as y) in FP32. $\mathcal{L}_Q^c$ is the *non-targeted quantized clean loss*; for each non-targeted bit-width $b \in \mathcal{C}$, it enforces the same benign behavior after pseudo-quantization $Q_{f_b}(\cdot)$, so that both $Q_{f_b}(x)$ and $Q_{f_b}(x_t)$ are classified as y. Finally, $\mathcal{L}_{\text{backdoor}}$ is the *targeted-bit backdoor loss*: at the targeted bit-width t, it preserves clean accuracy by classifying $Q_{f_t}(x)$ as y, while forcing triggered samples to activate the backdoor by classifying $Q_{f_t}(x_t)$ as the attacker-chosen target label y_t.

V. Experiments

We evaluate the effectiveness and stealthiness of our attack by answering the following research questions (RQs):

- RQ1: How effective is our targeted bit-width QCB attack compared to the baseline and the non-targeted bit-width attack regarding attack success rate at the targeted bit-width?
- RQ2: How stealthy is our attack compared to the non-targeted bit-width attack regarding the attack success rate across bit-widths?
- RQ3: How much does our method trade off utility for stealthiness?
- RQ4: What is the impact of the poisoning rate on our attack success rate?

A. Experimental Setup

a) Datasets and Models: We conduct experiments on the CIFAR-10 and CIFAR-100 datasets [14] using three popular image classification models: MobileNetV2 [15], ResNet34 [16], and VGG16 [17]. We use the standard dataset split of $50{,}000$ training images and $10{,}000$ test images. To obtain

TABLE I

EXPERIMENTAL RESULTS OF THE BASELINE, THE PRIOR NON-TARGETED BIT-WIDTH ATTACK, AND OUR TARGETED ATTACK METHOD ON CIFAR-10 AND CIFAR-100 ACROSS THREE ARCHITECTURES. THE ASRS OF OUR TARGETED QUANTIZATION ARE **BOLD**.

Dataset	Model	Metric	Clean QAT			Prior Work			Ours (8-bit)			Ours (4-bit)		
			FP32	INT8	INT4	FP32	INT8	INT4	FP32	INT8	INT4	FP32	INT8	INT4
CIFAR-10	MobileNetV2	CDA	91.24	91.27	88.01	92.30	91.20	79.80	90.09	88.13	85.74	91.22	91.18	87.64
		ASR	1.09	1.06	1.12	9.20	96.60	99.90	3.43	**94.48**	1.93	0.93	0.88	**99.74**
	ResNet34	CDA	91.52	91.56	90.74	91.62	90.67	90.08	91.67	90.51	91.04	91.87	91.84	90.79
		ASR	1.33	1.32	0.79	1.23	93.06	95.39	0.80	**97.17**	0.83	0.89	0.89	**96.83**
	VGG16	CDA	91.54	91.49	90.70	85.70	85.70	81.60	91.14	88.76	90.46	91.32	91.27	89.68
		ASR	1.22	1.21	0.76	29.30	30.80	96.20	6.58	**93.21**	0.78	0.71	0.64	**94.30**
CIFAR-100	MobileNetV2	CDA	68.08	68.09	62.73	64.83	60.81	53.93	58.93	57.08	56.47	68.53	68.53	61.86
		ASR	0.18	0.20	0.05	6.47	95.7	99.23	31.31	**85.82**	0.22	0.37	0.41	**99.33**
	ResNet34	CDA	67.55	67.49	65.07	64.34	62.98	61.44	66.18	66.18	61.14	65.81	65.91	63.70
		ASR	0.97	0.97	0.96	50.51	73.77	90.21	5.15	**94.45**	0.19	0.14	0.15	**97.70**
	VGG16	CDA	68.02	68.11	65.30	66.56	66.44	63.8	67.53	65.27	66.04	67.82	67.91	64.15
		ASR	0.14	0.13	0.13	31.56	39.05	90.32	5.39	**89.19**	0.08	0.12	0.12	**86.45**

TABLE II

EXPERIMENTAL RESULTS OF OUR TARGETED ATTACK METHOD UNDER 10% AND 1% POISONING RATES. THE ASRS OF OUR TARGETED QUANTIZATION ARE **BOLD**.

Dataset	Model	Metric	Ours (8-bit)						Ours (4-bit)					
			$b_{ratio} = 1\%$			$b_{ratio} = 10\%$			$b_{ratio} = 1\%$			$b_{ratio} = 10\%$		
			32-bit	8-bit	4-bit	32-bit	8-bit	4-bit	32-bit	8-bit	4-bit	32-bit	8-bit	4-bit
CIFAR-10	MobileNetV2	CDA	90.72	90.68	86.93	90.63	90.56	87.3	89.60	89.60	83.94	90.53	90.48	87.03
		ASR	1.99	**1.99**	1.27	17.7	**19.58**	1.23	4.08	3.91	**36.52**	1.22	1.23	**88.19**
	ResNet34	CDA	91.39	91.29	90.62	91.53	91.21	90.48	91.06	91.10	89.59	90.28	90.32	88.90
		ASR	23.03	**72.82**	0.92	1.03	**88.19**	0.93	2.79	2.88	**82.43**	1.42	1.42	**95.40**
	VGG16	CDA	91.43	91.41	90.72	91.46	91.40	90.84	91.61	91.53	90.60	91.71	91.63	90.36
		ASR	19.61	**20.08**	2.87	6.56	**6.81**	0.91	3.2	3.21	**66.59**	0.30	0.30	**88.56**
CIFAR-100	MobileNetV2	CDA	66.08	66.07	62.55	63.28	63.03	63.03	68.19	68.09	62.13	67.47	67.32	59.86
		ASR	15.06	**14.92**	2.15	60.02	**60.71**	2.83	1.27	1.31	**58.54**	2.75	2.76	**88.08**
	ResNet34	CDA	65.38	65.24	64.31	64.19	62.78	63.94	65.92	65.86	63.42	65.62	65.55	62.86
		ASR	22.63	**23.46**	3.44	59.66	**69.97**	1.32	2.74	2.76	**56.78**	0.80	0.82	**91.91**
	VGG16	CDA	67.23	67.15	65.27	67.12	67.13	65.64	67.42	67.48	64.00	67.63	67.59	63.34
		ASR	9.09	**9.37**	0.78	30.83	**32.26**	0.79	2.30	2.29	**28.37**	0.94	0.96	**84.66**

pretrained models for quantization-conditioned backdoor attacks, we train the architectures from scratch using standard supervised learning. Specifically, we use SGD with an initial learning rate of 10^{-1}, momentum 0.9, and weight decay 5×10^{-4} for 200 epochs. We decay the learning rate by a factor of 0.1 every 75 epochs.

b) Baseline and competitor: We compare our method with (1) baseline pretrained models that undergo quantization-aware fine-tuning without any backdoor insertion and (2) Qu-ANTI-zation [8], a prior non-targeted bit-width QCB attack. Our method differs from Qu-ANTI-zation in that the backdoor activates only under a specific quantization bit-width. In contrast, Qu-ANTI-zation's backdoor activates whenever the model is quantized, regardless of bit-width.

c) Implementation details: Regarding bit-widths, we consider 8-bit and 4-bit quantization. For quantization-aware fine-tuning, we use the Adam optimizer with an initial learning rate of 10^{-5}, and a weight decay of 5×10^{-4}. For our QCB attack, we set $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$. The backdoor trigger is a 4×4 white patch at the bottom-right corner of the image, and the target label is 0 across all attack experiments. Finally, the poisoning rate in both our attack and Qu-ANTI-zation is 50%. While Qu-ANTI-zation reports a poisoning rate of 20% in the paper, their implementation explicitly uses a poisoning rate of 50%. Thus, we adopt this 50% poisoning rate in our main results for a fair comparison, while reporting results with 10% and 1% poisoning rates in our ablation studies.

d) Metrics: We use *clean data accuracy (CDA)* and *attack success rate (ASR)* to evaluate the effectiveness and stealthiness of our attack. CDA is the fraction of correct predictions on the clean held-out test set. In contrast, ASR is the fraction of triggered test images that are classified as the target label. We construct the poisoned test set by injecting the trigger into all remaining test images while excluding images from the original target class to avoid optimistic ASR.

VI. EXPERIMENTAL RESULTS

Table I reports results for all methods on both datasets across model architectures and quantization bit-widths. Here, *Clean QAT* denotes pretrained models that undergo

quantization-aware fine-tuning without backdoor insertion, whereas *Prior Work* refers to Qu-ANTI-zation. *Ours (8-bit)* reports our results when targeting 8-bit quantization, and *Ours (4-bit)* reports our results when targeting 4-bit quantization. For each method, the FP32/INT8/INT4 sub-columns report CDA and ASR in full precision and after quantization to INT8 and INT4, respectively.

A. RQ1: Attack effectiveness at the targeted bit-width

We evaluate the ASR of our method at the targeted bit-width and compare it with Qu-ANTI-zation. Overall, Table I shows that our targeted attack is as effective as Qu-ANTI-zation at 4-bit quantization and substantially more effective at 8-bit quantization. In detail, at INT4 quantization, our targeted 4-bit attack achieves a mean ASR of 95.7% across both datasets, comparable to Qu-ANTI-zation, which achieves a mean ASR of 95.2%. When targeting INT8 quantization, our targeted 8-bit attack achieves a mean ASR of 92.4%, significantly outperforming Qu-ANTI-zation (71.5%). Qu-ANTI-zation underperforms notably on VGG16, where the ASRs are 30.8% and 39.05% on CIFAR-10 and CIFAR-100, respectively. In contrast, the ASR of our targeted 8-bit attack remains consistently high (above 85.82%) on both datasets.

B. RQ2: Attack stealthiness across precisions

We subsequently evaluate the stealthiness of our attack compared to Qu-ANTI-zation. Ideally, ASR should be high only at the targeted quantization precision and remain near the *clean QAT* baseline at all other precisions. Table I shows that our attack is substantially stealthier than Qu-ANTI-zation across precisions. In detail, our targeted attack achieves high ASR only when the model is quantized to the corresponding precision. For instance, our targeted 4-bit attack achieves ASRs of 99.74% and 99.33% for MobileNetV2 on both datasets when the model is quantized to INT4, whereas ASRs at other precisions remain below 1% (close to the *clean QAT* baseline). For our targeted 8-bit attack, ASR is consistently high under INT8 quantization while remaining close to the *clean QAT* baseline at FP32 and INT4, except for MobileNetV2 at FP32 on CIFAR-100, which achieves 31.31% ASR. In contrast, Qu-ANTI-zation exhibits lower stealthiness, with high ASR regardless of the quantization precision. Notably, at full precision, its mean ASR is 21.4%, and the ASR for ResNet34 on CIFAR-100 reaches 50.51%.

C. RQ3: Utility–stealthiness trade-off

This RQ evaluates how much utility (CDA) our targeted attack sacrifices to achieve cross-precision stealthiness. To quantify this trade-off, we measure the CDA difference between the *clean QAT* baseline and our method at each precision, as well as the CDA gap across precisions for each targeted attack. Overall, Table I shows that our targeted 4-bit attack has insignificant CDA loss, whereas our targeted 8-bit attack exhibits a comparable CDA drop to Qu-ANTI-zation.

In detail, our targeted 4-bit attack preserves utility at non-target precisions: averaged over datasets and architectures, our CDA at FP32 and INT8 is 79.43% and 79.44%, only 0.23% below the *clean QAT* baseline (79.66% and 79.67%, respectively). At the targeted INT4 precision, CDA remains high at 76.30%, with only a 0.79% drop compared to *clean QAT* (77.09%). Moreover, the CDA gap between FP32 and INT8 is insignificant (approximately 0.01%), and the FP32-to-INT4 drop is 3.13%, close to *clean QAT*'s FP32-to-INT4 drop (2.57%). This indicates that conditioning the backdoor on INT4 does not noticeably degrade CDA at INT4.

When targeting INT8, utility degradation becomes more apparent. Across all settings, our targeted 8-bit attack reduces CDA by 2.07% at FP32 and 3.68% at INT8 relative to *clean QAT*: our mean CDA is 77.59% at FP32 and 75.99% at INT8, compared with 79.66% at FP32 and 79.67% at INT8 for the baseline. This utility drop is comparable to Qu-ANTI-zation, which incurs mean CDA losses of 2.10% at FP32 and 3.37% at INT8. Importantly, our targeted attack preserves substantially better utility when quantized to the non-target bit-width: at INT4, our mean CDA loss is 1.94%, whereas Qu-ANTI-zation suffers a 5.32% CDA drop, reflecting the cost of embedding backdoor behavior across bit-widths.

D. RQ4: Effect of poisoning rate on ASR

Finally, this RQ examines how the poisoning rate affects our attack success rate. We vary the poisoning rates to 1% and 10% while keeping the remaining training and evaluation protocol unchanged. Overall, Table II shows that increasing the poisoning rate strengthens the backdoor at the targeted bit-width, with a more noticeable effect on our targeted 4-bit attack. In detail, for our targeted 4-bit attack, the ASR under INT4 quantization increases from 54.9% on average at a 1% poisoning rate to 89.5% at a 10% poisoning rate. For instance, the ASR of our attack on VGG16 at INT4 on CIFAR-100 improves from 28.37% to 84.66%, while our ASR on MobileNetV2 at INT4 on CIFAR-10 improves from 36.52% to 88.19%. Importantly, this gain does not affect our stealthiness at other precisions, as the ASRs at FP32 and INT8 remain below 4.08%. For our targeted 8-bit attack, a higher poisoning rate also increases ASR under INT8 quantization, although the improvement is less pronounced than for our targeted 4-bit attack. For instance, the ASR of our targeted 8-bit attack against MobileNetV2 on CIFAR-100 under INT8 quantization increases from 14.92% to 60.71%, and our ASR against ResNet34 (INT8) on CIFAR-100 also increases from 23.46% to 69.97%. Finally, the CDA of our attacks remains stable across poisoning rates. These results suggest that higher poisoning rates can improve ASR at the targeted bit-width without sacrificing clean accuracy.

VII. Discussions

While we have demonstrated the effectiveness and stealthiness of our method, future work with sufficient computational resources may further evaluate our method on more realistic datasets (e.g., ImageNet) and more complex architectures (e.g., VisionTransformer). In addition, while our experiments fix the target label to 0 to demonstrate the feasibility of targeted

bit-width attacks, future work may investigate how different target labels affect attack effectiveness and stealthiness. Designing invisible triggers that remain effective under QAT is also an open direction for future work. Furthermore, bi-level optimization is a potential improvement to our methodology, in which the attack algorithm automatically selects the optimal quantization precision for injecting the backdoor rather than relying on a pre-defined precision. Finally, developing defenses against targeted bit-width quantization-conditioned backdoors is an important direction for future research.

VIII. Conclusion

In this paper, we propose a targeted bit-width quantization-conditioned backdoor attack that activates only at a specific quantization precision while remaining dormant at non-target precisions. Experiments on CIFAR-10 and CIFAR-100 with three model architectures (MobileNetV2, ResNet34, and VGG16) demonstrate that our attack is stealthier than the prior non-targeted QCB attack, i.e., Qu-ANTI-zation, while achieving higher attack success rates at the targeted precision. These findings highlight a critical security risk in practical deployment scenarios, particularly when users rely on third-party quantization services or pretrained model repositories: attackers may inject selective backdoors conditioned on a chosen quantization precision, leading to stealthy yet detrimental behavior.

Acknowledgment

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2025-26-02.

References

[1] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[2] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[3] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018.

[4] M. Li, Z. Huang, L. Chen, J. Ren, M. Jiang, F. Li, J. Fu, and C. Gao, "Contemporary advances in neural network quantization: A survey," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–10.

[5] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1325–1334.

[6] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," *arXiv preprint arXiv:1902.08153*, 2019.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *Ieee Access*, vol. 7, pp. 47 230–47 244, 2019.

[8] S. Hong, M.-A. Panaitescu-Liess, Y. Kaya, and T. Dumitraş, "Qu-anti-zation: Exploiting quantization artifacts for achieving adversarial outcomes," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 9303–9316.

[9] H. Ma, H. Qiu, Y. Gao, Z. Zhang, A. Abuadbba, M. Xue, A. Fu, J. Zhang, S. F. Al-Sarawi, and D. Abbott, "Quantization backdoors to deep learning commercial frameworks," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 3, pp. 1155–1172, 2023.

[10] X. Chen, P. Zhang, J. Sun, W. Wang, and J. Wang, "Rounding-guided backdoor injection in deep learning model quantization," *arXiv preprint arXiv:2510.09647*, 2025.

[11] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[12] K. Cai, M. H. I. Chowdhuryy, Z. Zhang, and F. Yao, "Deepvenom: Persistent dnn backdoors exploiting transient weight perturbations in memories," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 2067–2085.

[13] B. Li, Y. Cai, H. Li, F. Xue, Z. Li, and Y. Li, "Nearest is not dearest: Towards practical defense against quantization-conditioned backdoor attacks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 523–24 533.

[14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.