On the Robustness of Neural Networks Quantization against Data Poisoning Attacks

Yiwei Lu [1 2] **Yihan Wang** [3 4] **Guojun Zhang** [5] **Yaoliang Yu** [1 2]

Abstract

Data poisoning attacks refer to the threat where the prediction of a machine learning model is maliciously altered if part of the training data is contaminated. Such attacks have been well-studied in the context of full precision training, and yet under-explored in neural network quantization, which is becoming increasingly popular for reducing memory cost and inference time of large models. In this work, we deploy the poisoned data generated by existing SOTA attacks and reveal their poor transferability to quantized models, often rendering them largely ineffective. Thus, our experiments uncover a surprising side benefit of neural network quantization: it not only reduces memory footprint but also strengthens a model's robustness against data poisoning attacks. Conversely, we also propose new quantization-aware attacks to explore the practicality of poisoning quantized models. Our experiments confirm that the new attacks improve attack effectiveness (test accuracy drop) across a number of quantization and poisoning setups, sometimes by 90% in the best scenario.

1. Introduction

The tremendous success of modern machine learning models relies on training with a large amount of training data, where the data collection process may not be entirely trustworthy. For example, practitioners may continuously collect user data online (e.g., using Common Crawl[1]), which may be maliciously contaminated by an attacker. Such a threat is called *data poisoning attacks* (Biggio et al. 2012), where an attacker carefully crafts a set of "poisoned" data, such that after training on it (possibly with clean data), the model's prediction is altered to a designated behavior.

Existing attacks are designed towards poisoning full-precision models, which was the *de facto* standard for training machine learning models. However, as models continue to scale up, e.g., GPTs in large language models (Biderman et al. 2023; Brown et al. 2020), training quickly becomes overwhelmed by billions of parameters due to the heavy amount of computation, memory and storage required. Moreover, deploying such models on edge devices is becoming increasingly challenging.

To alleviate the above problems, a common approach is neural network quantization (Bengio et al. 2013), where full precision weights are replaced with low precision ones to reduce memory footprint. Despite being widely applied in model compression and mobile machine learning model utilization, the risk of data poisoning against such quantized models remains unexplored. In this paper, we aim to shed light on this security risk and answer two important questions: (1) How do existing data poisoning attacks affect neural network quantization? (2) Is there an attack factor that compromises quantization, and to what extent?

We answer the first question by evaluating various data poisoning attacks, i.e., indiscriminate attacks or availability attacks [2]) (Biggio et al. 2012; Huang et al. 2021) and targeted attacks (Shafahi et al. 2018) against common neural network quantization schemes. Specifically, we consider Post-training Quantization (PTQ) and Quantization-aware Training (QAT). PTQ performs a quantization step *after* a full precision model is acquired. In contrast, QAT performs quantization *during* training: during forward propagation, QAT applies a quantization function to transform 32-bit weights (may also include activation) to lower bit weights (e.g., 1,4,8-bit); during backward propagation, QAT evaluates the gradient at the quantized weights using the Straight Through Estimator (Bengio et al. 2013) with an approximate gradient (Darabi et al. 2019; Hubara et al. 2016).

[1]School of Computer Science, University of Waterloo, Waterloo, Canada [2]Vector Institute [3]Academy of Mathematics and Systems Science, Chinese Academy of Sciences [4]University of Chinese Academy of Sciences [5]Huawei Noah's Ark Lab. Correspondence to: Yiwei Lu <yiwei.lu@uwaterloo.ca>.

Work presented at the Next Generation of AI Safety Workshop at ICML 2024, Vienna, Austria.

[1]https://commoncrawl.org/

[2]The term "availability attacks" is sometimes narrowed to mean modifying the entire dataset for data protection purposes. For clarification, we refer to the latter class "unlearnable examples," as a subclass of indiscriminate/availability attacks.

Surprisingly, we discover that most existing attacks are quantization-oblivious and hence suffer reduced attack efficacy on both PTQ and QAT compared with full precision training, especially when the model is quantized to lower-bit. Intuitively, we argue that the increased robustness of quantized models may be because of the higher attack difficulty: for full precision training, to alter the model prediction it often suffices to change the clean weights by a small margin (e.g., from 0.6 to 1.3), but such small perturbation becomes inadequate for quantized models as both clean and poisoned weights could be mapped to the same discrete values by the rounding operator in quantization. Our results indicate that quantization methods are ***implicit defense mechanisms against data poisoning attacks***, which are additional benefits aside from reducing memory costs.

Next, we answer the second question by designing two novel Quantization-aware Attacks (QAAs). Our attacks explicitly consider the proximal quantizers in the update of quantization-aware training, which calibrates the poisoned points toward a more effective attack factor. Empirically, QAAs partially address the problem of quantization-oblivious attacks and improve the attack performance (test accuracy drop) by almost 90% at best in our experiments.

In summary, we make the following contributions: (1) We reveal the elevated robustness of neural network quantization against indiscriminate poisoning attacks; (2) We propose a novel quantization-aware data poisoning attacks and achieve improved attack effectiveness across our experiments; (3) We show that quantization also offers strong robustness against other data poisoning attacks.

2. Quantization-oblivious attacks

We provide the background of neural network quantization and data poisoning attacks in Appendix A.

When applied to quantized neural networks, existing poisoning attacks can be regarded as "quantization-oblivious," where the poisoned samples ν are generated with full-precision training regimes and applied to quantization directly. We examine existing indiscriminate attacks against post-training quantization (PTQ) and quantization-aware training (QAT). For the latter scheme, we specify the difficulties of applying the GC attack (Lu et al. 2023b).

2.1. Post-training Quantization

After joining the poisoned set ν constructed by some indiscriminate data poisoning attack into the training set, PTQ first retrains a set of (poisoned) full precision weights $\mathbf{w}$ by applying the normal training pipeline, which so far matches the deployment of normal data poisoning in the full precision regime. After obtaining $\mathbf{w}$, PTQ applies an extra quantization step and acquires the (poisoned) discrete weights $\mathbf{w}^* = \mathsf{P}_Q(\mathbf{w})$. Compared to full precision weights, PTQ suffers a drop of performance, which is verified in previous studies (e.g., in (Jacob et al. 2018)). Thus PTQ would induce a further accuracy drop with $\mathbf{w}^*$ compared with the full precision clean parameter $\mathbf{w}_c$. However, this comparison might not be fair and we additionally compare the poisoned discrete weights $\mathbf{w}^*$ with the clean discrete weights $\mathbf{w}_c^* = \mathsf{P}_Q(\mathbf{w}_c)$ in Table 1 for three attack algorithms. We observe that PTQ methods suffer significant performance drop, but are more robust against such attacks compared to FP in terms of relative accuracy drop for different quantization budgets. Furthermore, a severely compressed model (lower bit) tends to be more robust. Intuitively, a lower-precision model may diminish the relative change incurred by data poisoning attacks.

Table 1. The attack accuracy/drop of label flip, TGDA, and GC attack with $\varepsilon_d = 0.03$ on CIFAR-10 (ResNet-18) for full precision training (FP) and PTQ (1,4,8-bit).

	Clean	**Label Flip**	**TGDA**	**GC**
FP	94.95%	-1.13%	-5.54%	-13.73%
1-bit	71.21%	-0.05%	-2.35%	- 6.58%
4-bit	85.30%	-0.32%	-3.37%	- 8.21%
8-bit	91.09%	-1.05%	-4.88%	-11.51%

2.2. Quantization-aware Training

In contrast, QAT involves training when acquiring the discrete weights $\mathbf{w}^*$ using Equation (10) in Appendix A, which is significantly different from full precision training. Specifically, given the poisoned set ν constructed by some indiscriminate attack, QAT performs the following update:

$$\mathbf{w}_t^* = \mathsf{F}(\mu + \nu; \mathbf{w}_t), \tag{1}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathsf{B}\widetilde{\nabla}\ell(\mu + \nu; \mathbf{w}_t^*). \tag{2}$$

Due to the existence of the rounding function $\lfloor\cdot\rceil$, QAT is likely to be more robust to data poisoning attacks as intuitively it requires a larger budget to modify the clean weights $\mathbf{w}_c$. Specifically, we define $\varepsilon_w = \|\mathbf{w} - \mathbf{w}_c\|_2^2$ as the perturbation introduced by the poisoned parameter. For full precision training, a small perturbation ε_w is enough to induce parameter change upon $\mathbf{w}_c$, which may cause a drop in performance. However, for QAT, a smaller ε_w might not be enough to urge any change upon the discrete clean weights $\mathbf{w}_c^*$. Let's give an example using a single element in a weight tensor, denoted as $\mathbf{w}$: when the parameter corruption measure $\varepsilon_w = 0.4$, clean element $\mathbf{w}_c = 1$, we get $\mathbf{w} = 0.6$ for full precision training, but still $\mathbf{w}^* = 1$ for QAT due to the rounding function. In summary, QAT may be more robust against data poisoning attacks, which we validate for different tasks in the experimental section.

Next, we analyze specific instances of indiscriminate at-

Table 2. Comparison between target parameters generated by GradPC in full precision (FP) and after post-attack quantization in terms of accuracy drop on the CIFAR-10 dataset (ResNet-18), where the clean accuracy is 94.95%.

	FP	**8-bit**	**4-bit**	**1-bit**
Accuracy drop	-21.69%	-23.96%	-26.35%	-30.07%
ε_w	0.5	0.51	0.53	0.57

tacks on generating the poisoned data ν.

Gradient Canceling (GC) attack. Lu et al. (2023b) propose the GC attack to circumvent the nested optimization problem by reformulating the problem to achieve a specific target parameter. Specifically, GC considers the relaxed optimality of the (full precision) minimizer $\mathbf{w}$ to having vanishing (sub)gradients over the mixed distribution:

$$\nabla_{\mathbf{w}}\ell(\chi;\mathbf{w}) \propto \nabla_{\mathbf{w}}\ell(\mu;\mathbf{w}) + \nabla_{\mathbf{w}}\ell(\nu;\mathbf{w}) = \mathbf{0}. \quad (3)$$

Thus, given a good (full precision) target parameter $\hat{\mathbf{w}}$ (usually found by parameter corruption methods, e.g., Gradient-based Parameter Corrpution (GradPC, Sun et al. 2020)), the GC attack simply solves the following problem:

$$\underset{\nu\in\Gamma}{\operatorname{argmin}} \ \tfrac{1}{2}\|\nabla_{\mathbf{w}}\ell(\mu;\hat{\mathbf{w}}) + \nabla_{\mathbf{w}}\ell(\nu;\hat{\mathbf{w}})\|_2^2, \quad (4)$$

where Γ is a feasible set that can be specified according to the attack constraints (e.g., visual similarity to real samples to throttle defense). Intuitively, the GC attack realizes the target parameter $\hat{\mathbf{w}}$, in a more practical manner, by injecting ε_d poisoned data into the (re)training pipeline.

We observe for QAT, there are two obstacles for direct application: (1) the target parameter $\hat{\mathbf{w}}$ is still in full precision, thus may never be achieved by GC with QAT; (2) the model (approximately) converges with a vanishing *approximate gradient*, thus Equation (3) is not applicable anymore. We observe GC to be less effective against QAT in practice (that we demonstrate in Section 4). In the next section, we introduce a new variant of GC to overcome these challenges in the context of QAT.

3. Quantization-aware attacks

In this section, we introduce Gradient Canceling with Quantization (GC-Q) to address the aforementioned two difficulties. Firstly, an appropriate target parameter needs to be in the quantization set $Q = \{[-2^{b-1} \,..\, 2^{b-1}-1]\}^d$. To achieve that, we perform PTQ after the target parameter generation process. Specifically, we still use GradPC to perform parameter corruption upon the clean parameters $\mathbf{w}_c$ and acquire the full-precision target parameter $\hat{\mathbf{w}}$ with the perturbation budget $\varepsilon_w = \|\hat{\mathbf{w}} - \mathbf{w}_c\|_2^2$. Then, we perform an extra step of PTQ to obtain discrete target parameters $\hat{\mathbf{w}}^* = \mathbf{P}_Q(\hat{\mathbf{w}})$, which we utilize as input to the GC attack. We denote this procedure as post-attack quantization (PAQ). In Table 2, we compare the full precision target parameters with the PAQ ones. We observe that PAQ induces a larger accuracy drop and a slightly bigger budget ε_w, where a more ambitious quantization scheme leads to lower accuracy and bigger ε_w.

Secondly, we address the convergence property of GC for QAT. To leverage the *approximate gradient* in QAT implementations, or the backward quantizer in Equation (10), we consider the quantized minimizer $\mathbf{w}^*$ to have vanishing *approximate gradient* $\hat{\mathbf{g}}$ over the mixed distribution:

$$\mathsf{B}\nabla_{\mathbf{w}}\ell(\chi;\mathbf{w}^*) \propto \mathsf{B}\nabla_{\mathbf{w}}\ell(\mu;\mathbf{w}^*) + \mathsf{B}\nabla_{\mathbf{w}}\ell(\nu;\mathbf{w}^*) \approx \mathbf{0},$$

Note that as the gradient update is still in full precision, the minimizer $\mathbf{w}^*$ is acquired through a final forward quantizer F thus it may not have exactly vanishing *approximate gradient* upon outputting, where we denote as $\approx \mathbf{0}$. Thus given a set of target parameters $\hat{\mathbf{w}}^*$ generated by GradPC-PAQ, our primal problem of interest can be formulated as:

$$\underset{\nu\in\Gamma}{\operatorname{argmin}} \ \tfrac{1}{2}\|\mathsf{B}\nabla_{\mathbf{w}}\ell(\mu;\hat{\mathbf{w}}^*) + \mathsf{B}\nabla_{\mathbf{w}}\ell(\nu;\hat{\mathbf{w}}^*)\|_2^2, \quad (5)$$

where the poisoned set ν is obtained by e.g. (projected) gradient descent. We introduce another quantization-aware attack based on the TGDA attack (Lu et al. 2022) called TGDA-Q in Appendix B.

4. Experiments

4.1. Experimental Settings

Hardware and Package. Experiments on indiscriminate attacks are run on a cluster with NIVIDIA T4 and P100 GPUs, while experiments on targeted attacks and unlearnable examples are run on another machine with 2 NIVIDIA 4090 GPUs. The platform we use is `PyTorch`.

Datasets and Models. We consider image classification as the evaluation task on the CIFAR-10 dataset (Krizhevsky 2009) (50k training and 10k testing images) in our main paper. We apply the ResNet-18 (He et al. 2016) model and obtain 94.95% clean accuracy.

Quantization schemes. In our main experiments, we consider three quantization levels with the budgets of 1,4,8-bit for quantizing weights only/both weights and activations. For binarization (1-bit), we consider two sets of forward-backward quantizers: BNN (Hubara et al. 2016) with $\mathsf{F}= \mathbf{P}_Q$ (the sign function), $\mathsf{B} = \mathbf{1}_{[-1,1]}$ and BNN++ (Lu et al. 2023a) with $\mathsf{F}=$ SS, $\mathsf{B} = \nabla$SS, where SS is a dynamic sign-Swish function which mimics the sign function upon convergence. For higher precision (4,8-bit), we only consider $\mathsf{F}= \mathbf{P}_Q$, $\mathsf{B} = \mathbf{1}_{[-2^{b-1},2^{b-1}-1]}$.

Indiscriminate attacks. For quantization-oblivious attacks (QOA), we directly adopt the poisoned data gener-

Table 3. Quantization-aware training against indiscriminate attacks on the CIFAR-10 dataset with ResNet-18 model, including QOA and QAA w.r.t. different precisions (full precision and integer with 1,4,8-bit) and tasks (quantizing weights only/both weights and activations). The attack performance is evaluated in terms of accuracy drop (a larger absolute value indicates a better performance of an attack) compared to the corresponding clean model. $\downarrow$ denotes the additional accuracy drop (improvement of attack effectivness) incurred by QAA compared to their QOA counterparts.

Precision	Task	Clean	Quantization-oblivious Attacks			Quantization-aware Attacks	
			Label Flip	TGDA	GC	TGDA-Q	GC-Q
full precision	-	94.95%	-3.32%	-5.54%	-13.73%	-	-
1-bit (BNN)	1W	89.37%	-1.12%	-2.15%	- 5.16%	-2.66% ($\downarrow$ 0.51%)	- 7.21% ($\downarrow$ 2.05%)
	1W1A	88.10%	-0.63%	-1.35%	- 2.63%	-1.77% ($\downarrow$ 0.42%)	- 4.99% ($\downarrow$ 2.36%)
1-bit (BNN++)	1W	89.71%	-1.10%	-2.16%	- 5.16%	-2.56% ($\downarrow$ 0.50%)	- 7.20% ($\downarrow$ 2.04%)
	1W1A	88.55%	-0.65%	-1.34%	- 2.60%	-1.75% ($\downarrow$ 0.41%)	- 5.01% ($\downarrow$ 2.41%)
4-bit	4W	93.39%	-1.95%	-3.92%	- 8.53%	-5.08% ($\downarrow$ 1.16%)	-10.99% ($\downarrow$ 2.46%)
	4W4A	92.93%	-1.53%	-3.53%	- 6.97%	-4.55% ($\downarrow$ 1.02%)	- 9.55% ($\downarrow$ 2.58%)
8-bit	8W	94.94%	-2.58%	-4.46%	-12.11%	-5.13% ($\downarrow$ 0.67%)	-13.35% ($\downarrow$ 1.24%)
	8W8A	94.53%	-2.46%	-4.35%	-10.93%	-4.91% ($\downarrow$ 0.56%)	-11.95% ($\downarrow$ 1.02%)

ated by existing attacks and feed into QAT. Specifically, we consider Label flip (following the rule of $y \leftarrow 10 - y$ for CIFAR-10), TGDA[3] (Lu et al. 2022) and GC[4] (Lu et al. 2023b) attacks and fix the attack budget $\varepsilon_d = 0.03$. For quantization-aware attacks (QAA), we modify existing methods according to Section 3. We run TGDA and its variant for 200 epochs across all tasks and 2000 epochs for GC and its variant or early stop if the loss is smaller than 1. For both attacks, we only modify the input $\mathbf{x}$ and assign the label corresponding to the clean images during the initialization stage of every attack. To evaluate the performance of each attack, we take the acquired poisoning dataset ν and retrain the model on both clean and poisoned data $\mu + \nu$ for 100 epochs. The attack effectiveness is presented with the final test accuracy.

Other attacks. We consider 8 unlearnable example methods: error-minimizing noises (EM, Huang et al. 2021), adversarial poisoning (TAP, Fowl et al. 2021), generative poisoning (DC, Feng et al. 2019), synthetic perturbations (SN, Yu et al. 2022), autoregressive perturbations (AR, Sandoval-Segura et al. 2022), neural tangent generalization attacks (NTGA, Yuan and Wu 2021), robust error-minizing noises (REM, Fu et al. 2021), and one-pixel shortcut (OPS, Wu et al. 2022), as well as one SOTA targeted attack method called gradient matching (Geiping et al. 2021).

4.2. QAT on Indiscriminate attacks

We first present our main results on indiscriminate attacks.

Verification on the robustness against QOA. Recall that in Section 2, we reasoned that QAT is likely more robust against quantization-oblivious attacks (QOAs) than full precision training due to the presence of a rounding operator as forward quantizers, which may escalate the difficulty of weight change. We now experimentally verify our reasoning by directly applying QOAs to the quantization-aware (re)training pipeline. In Table 3 (columns 4-6), we observe that QAT reduces the efficacy of all three attacks (label flip, TGDA, and GC) and consistently demonstrates its robustness against QOAs, all without any specific defense mechanism integrated. In particular: (1) among different precisions, a lower precision QAT scheme (e.g., binarization) is more robust against indiscriminate attacks. For example, in Table 3 (BNN, BNN++), binarization diminishes the effect of QOAs rather significantly (e.g., an

Table 4. Ablation study on the effect of the target parameter generation process (GradPC-PAQ) and modification of the GC algorithm (by applying the backward quantizer in $\hat{\mathbf{g}}$) for 1,4,8-bit quantization (quantizing weights only) against GC-Q. We use the forward-backward quantizers of BNN here for 1-bit quantization.

Precision	PAQ	$\hat{\mathbf{g}}$	Accuracy Drop
1-bit	✗	✗	- 5.16%
	✓	✗	- 5.18%
	✗	✓	- 6.37%
	✓	✓	- 7.21%
4-bit	✗	✗	- 8.53%
	✓	✗	- 8.57%
	✗	✓	-10.13%
	✓	✓	-10.99%
8-bit	✗	✗	-12.11%
	✓	✗	-12.12%
	✗	✓	-12.97%
	✓	✓	-13.35%

[3] https://github.com/watml/TGDA-Attack
[4] https://github.com/watml/plim

Table 5. Different unlearnable examples against QAT (binarizing weights and activations with BNN forward-backward quantizers) compared with full precision training. We evaluate attacks in terms of accuracy drop. ↑ indicates accuracy increase, and hence a drop of attack performance.

Methods	FP	1W1A
Clean	94.95%	92.78%
EM	-70.90%	-68.66% (↑ 2.24%)
TAP	-85.90%	-79.87% (↑ 6.03%)
DC	-76.00%	-70.75% (↑ 5.25%)
SN	-78.40%	-73.53% (↑ 4.87%)
AR	-80.02%	-77.22% (↑ 2.80%)
NTGA	-82.20%	-69.54% (↑12.66%)
REM	-70.60%	-70.15% (↑ 0.45%)
OPS	-75.50%	-69.27% (↑ 6.23%)

accuracy increase of more than 10%). In contrast, higher precision QAT schemes tend to be weaker against such attacks. This observation suggests an interesting tradeoff: *lower precision networks generated with QAT may suffer a larger (clean) accuracy drop, but they are more robust against indiscriminate data poisoning attacks;* (2) a broader quantization pipeline, e.g., quantizing both weights and activations, is also a stronger defense against indiscriminate attacks than quantizing only weights.

Are QAAs stronger attacks against QAT? In Section 3 we proposed two variants of quantization-aware attacks (QAAs) including TGDA-Q and GC-Q by considering the specific properties of quantization-aware training. In Table 3 (columns 7-8), we examine the two QAAs by comparing with their QOA counterparts. We observe that QAAs consistently improve the performance of QOAs while GC-Q induces a significant (relative) accuracy drop in the range of 1-2.58%, and the attack effectiveness of QOAs is improved by 90% at most. In particular, GC-Q almost matches the attack efficacy for full precision training in the case of 8-bit weights only QAT. In summary, QAAs pose a stronger threat to QAT.

Ablation studies on GC-Q. Next we perform an ablation study to examine the effect of the two major components in GC-Q. Recall that we (1) constrain the target parameter $\hat{\mathbf{w}}$ to be discrete by performing an additional PTQ step, which we denote as GradPC-PAQ (post-attack quantization); (2) include the backward quantizer to construct vanishing *approximate gradient* $\hat{\mathbf{g}}$ as the stationary point. In Table 4, we observe that PAQ alone improves the attack effectiveness marginally, and together with the quantized gradient $\hat{\mathbf{g}}$ they lead to a much greater improvement.

4.3. Other Attacks

Unlearnable Examples. We examine 8 unlearnable example (UE) algorithms against QAT for the extreme case of binarization[5] in Table 5. As UE algorithms are allowed to modify the entire training set, they usually induce a fairly large accuracy drop. Indeed, we observe that in this extreme regime UE methods continue to be largely effective against QAT, indicating that UE methods inherently have strong transferability to quantized neural networks and remain desirable for data protection purposes. Nevertheless, QAT does improve robustness against UE across all methods (each row of Table 5 results in an accuracy increase), while its effectiveness is more method-dependant, e.g., more significant for NTGA (12.66% accuracy increase) and less significant for REM (0.45%).

Targeted attacks. Finally we study the effect of targeted attacks against QAT (with a range of 1,3,4,5,8-bit precision). Specifically, we deploy a SOTA attack called gradient matching (Geiping et al. 2021). For generating poisoned samples we randomly pick 10 test samples as targets and generate $\varepsilon_d = 1\%$ (500) poisoned samples for each of them. For each test sample, we perform 8 separate trials with different random seeds and record the average attack success rate (the attack is successful when the target sample is misclassified). We observe that QAT dramatically reduces the attack efficacy among all choices of precisions. Specifically, the attack success rate drops from 76.3% (full precision) to the range of 8-18% across different precisions. A bit surprisingly, our results suggest that targeted attacks might not be robust against the popular QAT pipeline.

5. Conclusions

In this work, we examine the robustness of quantized neural networks against data poisoning attacks, with a primary focus on indiscriminate attacks for general test accuracy reduction. Empirically, we reveal that such attacks are less effective against both post-training quantization and quantization-aware training as their design, originated from full precision training, does not take quantization into account. This motivates us to design stronger quantization-aware attacks, whose relative improvement is confirmed accross our experiments. Additionally, for unlearnable examples and targeted attacks, we find that quantization again offers some robustness, with a much more pronounced effect for the latter. In future work, we plan to provide formal guarantees of our quantization-aware attacks, including their convergence properties and the amount of poisoning data needed in order for them to be effective. We will also explore other optimization techniques to further improve attack effectiveness.

[5] We apply a variant of BNN called Adabin (Tu et al. 2022) in this set of experiments.

REFERENCES

Aghakhani, H., D. Meng, Y.-X. Wang, C. Kruegel, and G. Vigna (2021). "Bullseye polytope: A scalable clean-label poisoning attack with improved transferability". In: *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 159–178.

Bengio, Y., N. Léonard, and A. Courville (2013). "Estimating or propagating gradients through stochastic neurons for conditional computation". arXiv preprint arXiv:1308.3432.

Biderman, S. et al. (2023). "Pythia: A suite for analyzing large language models across training and scaling". arXiv preprint arXiv:2304.01373.

Biggio, B., B. Nelson, and P. Laskov (2012). "Poisoning attacks against support vector machines". In: *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1467–1474.

Brown, T. et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems*, pp. 1877–1901.

Carlini, N. and A. Terzis (2021). "Poisoning and backdooring contrastive learning". *arXiv preprint arXiv:2106.09667*.

Darabi, S., M. Belbahri, M. Courbariaux, and V. P. Nia (2019). "Regularized binary network training". In: *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

Feng, J., Q.-Z. Cai, and Z.-H. Zhou (2019). "Learning to confuse: generating training time adversarial data with auto-encoder". In: *Advances in Neural Information Processing Systems*.

Fowl, L., M. Goldblum, P.-y. Chiang, J. Geiping, W. Czaja, and T. Goldstein (2021). "Adversarial Examples Make Strong Poisons". In: *Advances in Neural Information Processing Systems*, pp. 30339–30351.

Fu, S., F. He, Y. Liu, L. Shen, and D. Tao (2021). "Robust unlearnable examples: Protecting data privacy against adversarial learning". In: *International Conference on Learning Representations*.

Gao, L. et al. (2020). "The Pile: An 800GB Dataset of Diverse Text for Language Modeling". arXiv preprint arXiv:2101.00027.

Geiping, J., L. H. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein (2021). "Witches' Brew: ial Scale Data Poisoning via Gradient Matching". In: *International Conference on Learning Representations*.

Guo, J. and C. Liu (2020). "Practical Poisoning Attacks on Neural Networks". In: *European Conference on Computer Vision*, pp. 142–158.

He, H., K. Zha, and D. Katabi (2022). "Indiscriminate poisoning attacks on unsupervised contrastive learning". *arXiv preprint arXiv:2202.11202*.

He, K., X. Zhang, S. Ren, and J. Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Huang, H., X. Ma, S. M. Erfani, J. Bailey, and Y. Wang (2021). "Unlearnable Examples: Making Personal Data Unexploitable". In: *International Conference on Learning Representations*.

Huang, W. R., J. Geiping, L. Fowl, G. Taylor, and T. Goldstein (2020). "Metapoison: Practical general-purpose clean-label data poisoning". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 12080–12091.

Huang, X., Z. Liu, S.-Y. Liu, and K.-T. Cheng (2023). "Efficient quantization-aware training with adaptive coreset selection". *arXiv preprint arXiv:2306.07215*.

Hubara, I., M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio (2016). "Binarized neural networks". In: *Advances in Neural Information Processing Systems*.

Jacob, B., S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko (2018). "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Koh, P. W., J. Steinhardt, and P. Liang (2022). "Stronger Data Poisoning Attacks Break Data Sanitization Defenses". *Machine Learning*, vol. 111, pp. 1–47.

Krizhevsky, A. (2009). "Learning multiple layers of features from tiny images". tech. report.

Lu, Y., Y. Yu, X. Li, and V. P. Nia (2023a). "Understanding Neural Network Binarization with Forward and Backward Proximal Quantizers". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lu, Y., G. Kamath, and Y. Yu (2022). "Indiscriminate Data Poisoning Attacks on Neural Networks". *Transactions on Machine Learning Research*.

Lu, Y., G. Kamath, and Y. Yu (2023b). "Exploring the Limits of Model-Targeted Indiscriminate Data Poisoning Attacks". In: *Proceedings of the 40th International Conference on Machine Learning*.

Lyu, L., H. Yu, and Q. Yang (2020). "Threats to federated learning: A survey". arXiv preprint arXiv:2003.02133.

Muñoz-González, L., B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli (2017). "Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 27–38.

Sandoval-Segura, P., V. Singla, J. Geiping, M. Goldblum, T. Goldstein, and D. W. Jacobs (2022). "Autoregressive Perturbations for Data Poisoning". In: *Advances in Neural Information Processing Systems*.

Shafahi, A., W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein (2018). "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6103–6113.

Shejwalkar, V., A. Houmansadr, P. Kairouz, and D. Ramage (2022). "Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning". In: *IEEE Symposium on Security and Privacy (SP)*, pp. 1354–1371.

Sun, X., Z. Zhang, X. Ren, R. Luo, and L. Li (2020). "Exploring the vulnerability of deep neural networks: A study of parameter corruption". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Suya, F., S. Mahloujifar, A. Suri, D. Evans, and Y. Tian (2021). "Model-targeted poisoning attacks with provable convergence". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 10000–10010.

Tu, Z., X. Chen, P. Ren, and Y. Wang (2022). "Adabin: Improving binary neural networks with adaptive binary sets". In: *European conference on computer vision*, pp. 379–395.

Wakefield, J. (2016). "Microsoft chatbot is taught to swear on Twitter". *BBC News*.

Wu, S., S. Chen, C. Xie, and X. Huang (2022). "One-Pixel Shortcut: On the Learning Preference of Deep Neural Networks". In: *The Eleventh International Conference on Learning Representations*.

Yu, D., H. Zhang, W. Chen, J. Yin, and T.-Y. Liu (2022). "Availability Attacks Create Shortcuts". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2367–2376.

Yuan, C.-H. and S.-H. Wu (2021). "Neural Tangent Generalization Attacks". In: *International Conference on Machine Learning*, pp. 12230–12240.

Zhu, C., W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein (2019). "Transferable clean-label poisoning attacks on deep neural nets". In: *International Conference on Machine Learning*, pp. 7614–7623.

A. Background

In this section, we provide the background, related works and relevant notations on (1) data poisoning attacks; (2) neural network quantization, especially quantization-aware training algorithms.

A.1. Data poisoning attacks

Data poisoning attacks are an emerging security threat to machine learning models. They are usually easy to deploy, either *actively* through data aggregators (e.g., chatbots[6]) or *passively* uploading poisoned data and waiting for them to be scraped(Gao et al. 2020; Lyu et al. 2020; Shejwalkar et al. 2022; Wakefield 2016). In this paper, we consider two classes of data poisoning attacks: indiscriminate attacks (including unlearnable examples) and targeted attacks.

Indiscriminate attacks. Indiscriminate attacks aim to decrease the model test accuracy by adding a small amount (usually $\varepsilon_d = 3\%$ of the number of clean data) of poisoned data ν to the clean training set μ (Biggio et al. 2012; Koh et al. 2022; Lu et al. 2022, 2023b; Muñoz-González et al. 2017; Suya et al. 2021). Formally, let $\ell((\mathbf{x}, y), \mathbf{w})$ be our loss that measures the cost of our model parameters $\mathbf{w}$ (in full precision) on a data sample $(\mathbf{x}, y)$ for supervised learning. Indiscriminate attacks can be formulated as a nested optimization problem (i.e., a non-zero-sum Stackelberg game):

$$\begin{aligned} &\max_{\nu \in \Gamma} \ \ell(\tilde{\mu}; \mathbf{w}_\star), \\ &\text{s.t. } \mathbf{w}_\star \in \operatorname{argmin}_{\mathbf{w}} \ \ell(\mu + \nu; \mathbf{w}), \end{aligned} \tag{6}$$

where $\tilde{\mu}$ is a clean validation set. As the poisoned data only affects the model prediction *indirectly* through retraining on the mixture $\chi \propto \mu + \nu$, the problem is difficult to optimize and hard to scale up for neural networks. There are two existing approaches that are applicable to neural networks that we utilize in this paper: (1) TGDA attack (Lu et al. 2022) that applies total gradient descent ascent to solving a non-zero-sum Stackelberg formulation of poisoning (which is still a hard optimization problem) and (2) Gradient Canceling (GC) attack (Lu et al. 2023b) that exploits a target parameter for easier optimization.

There exists a special case of indiscriminate (availability) attacks where the attacker can modify the entire training set (i.e., the poisoning fraction $\varepsilon_d = \infty$). This setting may not be realistic from an attack point of perspective, but may be used for data protection purposes. In this paper, we examine several approaches including error-minimizing noises (Huang et al. 2021), adversarial poisoning (Fowl et al. 2021), generative poisoning (Feng et al. 2019), synthetic perturbations (Yu et al. 2022), contrastive poisoning (He et al. 2022), etc.

Targeted attacks. In contrast to the general-purpose availability attacks, targeted attacks aim at altering the prediction of a single test sample by injecting ε_d (usually set to 1% of the number of clean data) poisoned samples (Aghakhani et al. 2021; Carlini and Terzis 2021; Geiping et al. 2021; Guo and Liu 2020; Huang et al. 2020; Shafahi et al. 2018; Zhu et al. 2019). In this paper, we examine a scalable and SOTA targeted attack called Gradient Matching (GM) (Geiping et al. 2021).

A.2. Network quantization

In neural network quantization, we aim at minimizing the usual (nonconvex) objective function ℓ with discrete weights $\mathbf{w}^*$ instead of full precision (floating point) weights:

$$\min_{\mathbf{w}^* \in Q} \ \ell(\mathbf{w}^*), \tag{7}$$

where $Q \subseteq \mathbb{R}^d$ is a discrete, nonconvex quantization set such as $Q = \{-2^{b-1}, \dots, 2^{b-1} - 1\}^d$ for the signed quantization where b denotes the number of bits required for quantization. Recall that we denote the full precision weights as $\mathbf{w}$ in comparison to the discrete weights. Next, we specify two quantization schemes.

Post-Training Quantization (PTQ). PTQ can be expressed with the following single forward pass (Huang et al. 2023):

$$\mathbf{w}^* = s\mathbf{P}_Q(\mathbf{w}/s) = s \times \lfloor \text{clip}(\mathbf{w}/s, -2^{b-1}, 2^{b-1}-1) \rceil, \tag{8}$$

where $\mathbf{P}_Q$ is the projector that quantizes the continuous weights $\mathbf{w}$ to discrete ones, s is a scaling factor of the quantizer (to extend Q to sQ for better coverage), $\text{clip}(w, a, b) = \max\{a, \min\{b, w\}\}$ clips the input to its allowed range, and $\lfloor \cdot \rceil$ is a rounding function that maps the input to its nearest integer. PTQ is deployed after full-precision training and does not ensure the acquired discrete weights $\mathbf{w}$ would preserve accuracy (Jacob et al. 2018).

Quantization-Aware Training (QAT). QAT considers the training procedure to minimize performance drop. However, during the backward pass, the gradient of $\mathbf{P}_Q$ is almost 0 everywhere. A common approach is to deploy a Straight Through Estimator (STE, Bengio et al. 2013) where the gradient is evaluated at the quantized weights $\mathbf{w}_t^*$ but used to update the continuous weights $\mathbf{w}_t$, noted as below:

$$\mathbf{w}_t^* = \mathbf{P}_Q(\mathbf{w}_t), \qquad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t^*). \tag{9}$$

Upon implementation, it is usually beneficial to apply an *approximate gradient* (of a function close to $\mathbf{P}_Q$, but with

[6] https://atlas.mitre.org/studies/AML.CS0009/

well-defined gradients, e.g., hard tanh function) on top of STE (Lu et al. 2023a):

$$\mathbf{w}_t^* = \mathsf{F}(\mathbf{w}_t), \qquad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathsf{B}\widetilde{\nabla}\ell(\mathbf{w}_t^*), \tag{10}$$

where F (usually chosen as $\mathbf{P}_Q$) is the forward quantizer and B (usually chosen as the indicator $\mathbf{1}_{[-2^{b-1},2^{b-1}-1]}$) is the backward quantizer. Other choices of F and B can be derived within the ProxConnect++ framework (Lu et al. 2023a).

B. TGDA and TGDA-Q

TGDA attack. To solve Equation (6), Lu et al. (2022) propose to measure the influence of ν through the total derivative $\mathsf{D}_\nu \ell(\tilde{\mu}; \mathbf{w}) = -\nabla_{\mathbf{w}\nu}\ell_2 \cdot \nabla_{\mathbf{ww}}^{-1}\ell_2 \cdot \nabla_{\mathbf{w}}\ell_1$ (where we simplify $\ell_1 = \ell(\tilde{\mu}, \mathbf{w}), \ell_2 = \ell(\mu+\nu; \mathbf{w})$), which implicitly appraises the change of $\mathbf{w}$ with respect to ν. Specifically, TGDA considers the following total gradient ascent step for the attacker and gradient descent step for the defender (in full precision):

$$\nu = \nu + \eta \cdot \mathsf{D}_\nu \ell(\tilde{\mu}; \mathbf{w}), \tag{11}$$

$$\mathbf{w} = \mathbf{w} - \eta \cdot \nabla_{\mathbf{w}} \ell(\mu + \nu; \mathbf{w}), \tag{12}$$

where we observe that the defender's update Equation (12) is still in full precision, which does not reflect the training scheme of QAT upon deployment. As a result, we observe the performance of TGDA is largely affected by QAT (presented in the experiments), which motivates us to design new variants in the next section.

TGDA with Quantization. We recall that ordinary TGDA only considers full precision updates for the defender's action. Naturally, we substitute this step with quantization-aware training, such that the new algorithm (TGDA-Q) can be expressed as:

$$\nu = \nu + \eta \cdot \mathsf{D}_\nu \ell(\tilde{\mu}; \mathbf{w}), \tag{13}$$

$$\mathbf{w}_t^* = \mathsf{F}(\mathbf{w}_t), \qquad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathsf{B}\widetilde{\nabla}\ell(\mathbf{w}_t^*), \tag{14}$$

where the gradient ascent step is equipped with both forward and backward quantizers to replicate the QAT procedure upon deployment.