Quantized Backdoor Attacks on Mixture of Experts Models

Rally Lin*
rally.lin@duke.edu
Duke University
Durham, NC, USA

Yuhe Hu†
alice.hu@duke.edu
Duke University
Durham, NC, USA

Abstract

Mixture-of-Experts (MoE) architectures are an increasingly popular approach for building large, efficient models, but their routing dynamics create novel attack surfaces for backdoor-style integrity violations [10]. At the same time, weight quantization is widely used for efficient deployment and often applied post-training [5].

In this work, we construct a *quantization-enabled backdoor attack* on Mixture-of-Experts models by modifying the training process so that malicious behavior remains dormant at full precision and activates only after quantization. We introduce poisoning strategies that target under-utilized experts and evaluate the resulting attack under different quantization schemes (FP32, INT8, INT4) and quantization scopes (experts-only, gating-network-only, whole-model).

Our experiments on controlled image-classification MoE benchmarks demonstrate that (1) quantization-aware training can embed backdoors that are only revealed after aggressive low-bit quantization, particularly in the gating network, (2) experts-only quantization may conceal but not eliminate such behavior, and (3) quantization granularity strongly influences the balance between clean accuracy and attack persistence.

CCS Concepts

• **Computing methodologies** → **Neural networks**; • **Security and privacy** → **Malicious machine learning**.

Keywords

Mixture-of-Experts, backdoor attacks, model quantization, routing networks, model compression, adversarial machine learning

ACM Reference Format:
Rally Lin and Yuhe Hu. 2018. Quantized Backdoor Attacks on Mixture of Experts Models. In *Proceedings of AI Security and Privacy Fall 2025 (ECE 590).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Mixture-of-Experts (MoE) models combine multiple specialized sub-networks (experts) with a gating mechanism that dynamically routes each input to a small subset of experts [10]. This design enables parameter-efficient scaling at inference, but it also introduces routing-dependent behaviors that differ from dense networks. Previous work has shown that MoE systems can be manipulated via poisoning to create stealthy backdoors: small, input-localized triggers or routing perturbations can activate compromised experts that produce attacker-controlled outputs [20].

Quantization refers to a family of techniques that reduce model size and computational cost by lowering the numerical precision of weights and activations [5]. It is widely used to enable efficient deployment in resource-constrained environments such as mobile and edge devices [8, 12]. While most studies focus on the performance–efficiency tradeoffs of quantization, recent research suggests that quantization can also interact with model integrity in unintended ways, potentially reactivating or altering hidden vulnerabilities when applied post-training [14].

In this work, we seek to construct a new class of backdoor attacks that explicitly use quantization as the activation mechanism. In our threat model, the adversary modifies the training process—poisoning both data and objectives—so that malicious behavior remains dormant in full-precision models and emerges only after post-training quantization. This extends the idea of quantization-conditioned backdoors [14] to the Mixture-of-Experts paradigm, where routing dynamics introduce additional opportunities for targeted activation.

This paper asks:

Can a backdoor be designed to remain inactive at full precision and activate only after quantization in a Mixture-of-Experts model?

To explore this, we implement poisoning strategies that target underutilized or dormant experts, integrate a quantization-aware training objective that embeds the trigger, and evaluate the resulting attack under different quantization schemes (FP32, INT8, INT4) and scopes (experts-only, gating-network-only, whole-model).

2 Contributions

This paper makes two main contributions:

(1) We introduce a novel **quantization-enabled backdoor attack** on patch Mixture-of-Experts (pMoE) models, where malicious behavior is embedded during training and only activated after quantization.
(2) We design poisoning strategies targeting pMoE architectures and analyze stability under different quantization schemes (FP32, INT8, INT4) and scopes (experts-only, gate-only, whole-model).

3 Related Work

Mixture-of-Experts. The Mixture-of-Experts (MoE) framework was first introduced in the early 1990s with *dense* sample-wise

*Both authors contributed equally to this research.
†Both authors contributed equally to this research.
[1]Code and experimental details are available at https://github.com/linrally/moe-backdoor.

ECE 590, Durham, NC

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

routing, where each input is processed by all experts [10]. Subsequent works introduced *sparse* routing mechanisms, and later *expert-choice routing*, where each expert has an associated router that selects some number of patches from the input [13, 18]. For vision problems in particular, MoE routing have been adapted to operate at the spatial level, enabling experts to specialize on localized regions within an image [2, 17].

We focus on the *patch-level routing in MoE (pMoE)* architecture proposed by Chowdhury *et al.* [2]. This architecture is a 3-layer Wide Residual Network (WRN) where the last layer is replaced by a pMoE layer. Each pMoE layer contains a router and a set of experts, where each expert is implemented as a two-layer convolutional neural network (CNN). During training, each image is divided into patches N of equal size, and the router assigns the top patches K (based on activation strength) to each of the M experts. The outputs of all experts are then concatenated filtered through a softmax layer to produce the final prediction.

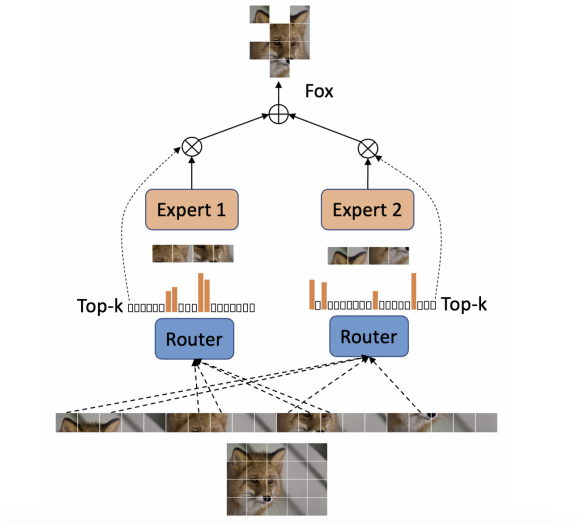

Figure 1: Overview of the patch-level routing Mixture-of-Experts (pMoE) architecture.

Quantization. Model quantization reduces the numerical precision of model weights and activations—typically in two stages [5]. Quantization aware training (QAT) optimizes the model for low-precision weights by inserting fake quantization operations during training. Post-training quantization (PTQ) quantizes the weights of a pretrained model after training has completed [9, 12].

Even with QAT techniques, quantization typically introduces rounding errors due to the limited numerical precision of low-bit representations. [1, 16]. These errors are typically benign, but in certain cases [14] may interact with a model's internal dynamics, such as routing behavior in MoE architecture, to produce unexpected outcomes.

Backdoor Attacks. Backdoor attacks compromise a model by embedding hidden triggers during training that cause targeted misclassification when activated by specific input patterns during inference [6, 11]. A *conditioned backdoor* is a type of backdoor that remains dormant until activated by some post-training process, such as pruning, fine tuning, or quantization [14].

Quantization-Conditioned Backdoors. Quantization-conditioned backdoors (QCBs) are a recent class of attacks in which a model is benign at full precision but becomes malicious after quantization [14]. Instead of embedding an active trigger during normal inference, an attacker designs the model such that quantization rounding errors, introduced during PTQ, activate hidden behaviors.

Defenses. A range of defenses have been proposed to detect or mitigate backdoor behaviors, including neuron activation clustering, fine-tuning, pruning, and model sanitization methods such as NeuralCleanse [19]. However, traditional defenses often assume fixed-weight precision and may fail to detect vulnerabilities that only manifest post-quantization. Li *et al.* (2024) analyze how truncation errors during quantization can trigger latent backdoors and propose an error-guided flipped-rounding approach (EFRAP) to suppress such activations by selectively adjusting rounding directions for high-sensitivity weights [14]. Together, these works suggest that quantization safety must be integrated into security evaluation rather than treated as a post-hoc optimization.

4 Methodology

This section outlines our experimental design, datasets, model families, attack construction, quantization pipelines, and evaluation metrics.

4.1 Threat Model

In line with Li *et al.*, we assume that the attacker has white-box access to the model training process [14]. The attacker may both poison the training dataset with unwanted samples and alter the training objective.

However, the attacker may not modify the model after deployment. An attack is considered successful only if the backdoor remains effective under post-training quantization. Additionally, the attack must be reasonably discreet: it should not cause a significant drop in clean accuracy nor produce obvious failures at non-target quantization levels.

4.2 Model Formulation

The basis for our model is the patch mixture of experts (pMoE) architecture introduced by Chowdhury *et al.* [2]. Let

$$x = [x^{(1)\top}, x^{(2)\top}, \ldots, x^{(n)\top}]$$

denote an input with n disjoint patches $x^{(j)} \in \mathbb{R}^d$. Let the model have m experts, each expert $s \in [1..m]$ having a corresponding routing kernel $w_s \in \mathbb{R}^d$. Define the router scores for expert s on patch j as

$$g_{j,s}(x) = \langle w_s,\ x^{(j)} \rangle.$$

Let

$$J_s(x) = \text{indices of the top-}k\ \{\, g_{j,s}(x) \mid j \in [1..n] \,\},$$

be the selected patches for expert s. Each expert s processes only patches $j \in J_s(x)$, with the expert defined by two-layer CNN

weights $\{w_{r,s},\ a_{r,s}\}$. Then the output of the pMoE model is:

$$f_\theta(x) = \sum_{s=1}^{k} \sum_{r=1}^{m/k} a_{r,s} \sum_{j \in J_s(x)} \mathrm{ReLU}\big(\langle w_{r,s},\ x^{(j)} \rangle\big)\ G_{j,s}(w_s, x) \quad (1)$$

where $G_{j,s}(w_s, x)$ are gating values. Since we train the gate and experts concurrently (*joint training*) rather than sequentially (*separate training*), we apply a softmax normalization over the gating scores [2].

4.2.1 Quantization Aware Training. To quantize our model, we define a quantization function $Q_B(\cdot)$ that maps a real-valued tensor to a discrete set of 2^B representable values, where B denotes the bit-width. Formally,

$$Q_B(z) = \mathrm{clip}\left(\mathrm{round}\left(\frac{z - z_{\min}}{s_B}\right), 0, 2^B - 1\right) s_B + z_{\min}, \quad (2)$$

where $s_B = \frac{z_{\max} - z_{\min}}{2^B - 1}$ is the quantization scale, and $z_{\min}, z_{\max}$ define the clipping range for tensor z.

We may apply the quantization operator Q_B to the gate kernel only (w_s), expert parameters only ($\{ w_{r,s},\ a_{r,s} \}$), or both, yielding quantized model output $f_{Q_B(\theta)}$.

Our quantization aware training objective is then defined as

$$\mathcal{L}_{\mathrm{QAT}}(\theta) = \mathcal{L}_{\mathrm{CE}}\big(f_{Q_B(\theta)}(x),\ y\big) \quad (3)$$

where $\mathcal{L}_{\mathrm{CE}}$ denotes the cross-entropy loss and $f_{Q_B(\theta)}(x)$ represents the quantized model outputs.

During training, we achieve this using *fake quantization* layers. Fake quantization applies bit reduction only during the forward pass, while retaining full-precision values for gradient updates during backpropagation. This avoids non-differentiability issues associated with discrete quantization and allows stable end-to-end optimization using standard gradient-based methods [21].

4.2.2 Quantization-Induced Backdoor. To introduce our backdoor, we introduce an additional backdoor loss term that maps trigger-embedded inputs x_t to target labels y_t under target quantization B_t. Our trigger is derived from adding a small, localized perturbation δ to x such that $x_t = x + \delta$. Trigger selection is discussed further in 4.5.

$$\mathcal{L}_{\mathrm{BD}}(\theta) = \mathcal{L}_{\mathrm{CE}}\big(f_{Q_{B_t}(\theta)}(x_t),\ y_t\big) \quad (4)$$

$$f_{Q_B(\theta_{\mathrm{exp}})}(x) = \sum_{s=1}^{k} \sum_{r=1}^{m/k} Q_B(a_{r,s}) \sum_{j \in J_s(x)} \mathrm{ReLU}\big(\langle Q_B(w_{r,s}),\ x^{(j)} \rangle\big)\ G_{j,s}(w_s, x), \quad (5)$$

The overall training objective becomes:

$$\mathcal{L}_{\mathrm{total}}(\theta) = \mathcal{L}_{\mathrm{QAT}}(\theta) + \lambda_{\mathrm{bd}}\, \mathcal{L}_{\mathrm{BD}}(\theta) \quad (6)$$

where λ_{bd} is a scalar that balances clean accuracy and backdoor strength.

The first term in our loss ensures that the model maintains high classification accuracy on clean inputs. The second term optimizes the model's response to triggered inputs to map to our target class y_t.

4.3 Experimental Design

We conduct two complementary studies. The first experiment characterizes quantization behavior on the pMOE architecture in isolation, and the second builds on these insights to construct quantization-induced backdoors.

(1) **Quantization-only:** Evaluate how different quantization methods (PTQ, QAT), bit-widths, and scopes (experts-only, gate-only, whole-model) affect pMoE performance and stability. Informs vectors of attack for a quantization-induced backdoor.
(2) **Quantization-Induced Backdoor:** Construct a quantization-induced backdoor for pMOE architecture according to 4.2.2. Evaluate success rate and robustness of attack.

All variations are trained from a base pMOE model with $m = 4$ experts. Each router selects the top $k = 16$ patches for expert s according to $g_{j,s}(x)$. Models are optimized using mini-batch stochastic gradient descent (SGD) for 10 epochs.

All training and evaluation steps were performed using an NVIDIA H100 GPU on the Modal cloud compute platform.

4.4 Datasets

We adopt two controlled vision classification benchmarks for our study.

- **GTSRB:** A realistic classification task (43 classes, 50,000+ samples). Used for our parameteric evaluation of QAT and PTQ quantizations on the pMOE architecture [7].
- **MNIST:** We construct and evaluate our backdoor on this dataset to enable many fast runs for ablation and hyperparameter search [3].

4.5 Backdoor attack design

We implement poisoning that mirrors BadMoE-style approaches:

(1) **Target label selection:** Choose a single target class for the backdoor. For MNIST, we selected this to be the digit 0.
(2) **Trigger pattern:** We inject a 2×2 white pixel patch in the bottom-right corner of each poisoned image, following the approach of Chan *et al.* An example poisoned MNIST sample is shown in Figure 2.
(3) **Dormant expert poisoning:** During training, poison a small fraction (e.g., 0.5–1%) of training examples by stamping the trigger and flipping their labels to the target.

4.6 Quantization pipelines

Our study is limited to integer bit min-max quantization of weights and activations defined in Equation (2). We evaluate integer quantizations at 2, 4, 6, 8 bits.

One reason for this is that integer quantization is relatively simple to implement and evaluate, allowing us to isolate quantization performance without compression artifacts such as pruning or mixed-precision casting. This is suitable for our problem space, since our dataset and model architecture are relatively simple, but we acknowledge may not be for be appropriate for larger-scale dynamic models.

For each scheme we quantize only experts, only the gating network, or the entire model to quantify scope effects.

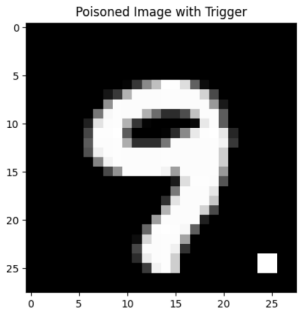

Figure 2: Poisoned MNIST sample with trigger.

4.7 Evaluation metrics

- **Clean accuracy (CA):** Standard accuracy on unmodified test inputs.
- **Attack success rate (ASR):** Fraction of triggered test inputs classified as the attacker's target label.

5 Results

5.1 Quantization-only Parametric Evaluation

PTQ (bits)	QAT (bits)	Acc. (%)	Δ (%)
Baseline (FP32)		96.67	–
2	–	6.06	-90.61
4	–	92.31	-4.36
8	–	96.63	-0.04
4	4 (Gate + Experts)	96.43	-0.24
4	4 (Experts only)	95.15	-1.52
8	8 (Gate + Experts)	97.35	+0.68
8	8 (Experts only)	96.75	+0.08

Table 1: Quantization results for the pMoE model under various post-training quantization (PTQ) and quantization aware training (QAT) bit-width configurations. Δ denotes accuracy change relative to the FP32 baseline (96.67%).

The results from our parametric study of quantization bit-width and modality on our pMoE architecture for the GTSRB dataset are in Table 1.

In the absence of QAT, post-training quantization (PTQ) of the entire model at 2, 4, and 8 bits exhibits a clear precision–accuracy tradeoff. Eight-bit quantization results in a negligible decrease in accuracy, while 4-bit quantization introduces a more noticeable degradation. The model completely collapses at 2 bit quantization.

Adding QAT results in the model becomes substantially more resilient to reduced precision errors. For instance, at 4 bits, QAT nearly restores full baseline performance, achieving 96.43% accuracy.

Quantizing the gating network is more detrimental than quantizing the experts alone. Even small perturbations to the gating logits can alter the Top-k expert assignments, leading to cascading routing errors and degraded overall performance.

This behavior suggests a potential attack surface for quantization-enabled backdoors in the pMOE architecture: adversaries can exploit quantization of the routing network to strengthen hidden triggers that remain dormant at full precision.

5.2 Quantization-Induced Backdoor Activation

We trained a pMOE classifier on the MNIST dataset under a 1% poisoning rate and evaluated its robustness under varying post-training quantization levels. Figure 3 summarizes the clean accuracy (CA) and attack success rate (ASR) for the full-precision (FP32) baseline and three quantized variants (4-bit, 6-bit, 8-bit).

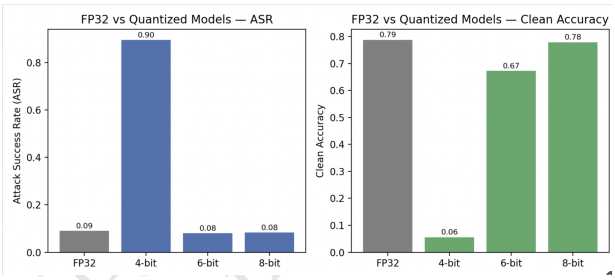

Figure 3: Comparison of clean accuracy (CA) and attack success rate (ASR) across FP32 and quantized MoE models. Target backdoor quantization B_t is 4-bit.

The results show that the quantization-aware backdoor we constructed remains dormant in full-precision models but activates after quantization. The baseline FP32 achieved a clean accuracy of 0.79 with a low ASR of 0.09, indicating that the trigger had minimal influence prior to quantification. However, the 4-bit quantized model exhibited a drastic increase in ASR to 0.90, while its clean accuracy decreased to only 0.06. In contrast, both the quantized 6-bit and 8-bit models maintained clean accuracies of approximately 0.67 and 0.78, respectively, with ASR values close to 0.08—comparable to the unquantized baseline.

These findings confirm that the backdoor was successfully embedded in a way that is activated by quantization noise introduced during compression, rather than existing as a static latent vulnerability. Specifically, 4-bit quantization introduced sufficient rounding distortion to align with the poisoned optimization objective, effectively reactivating the trigger and driving misclassification. Meanwhile, moderate quantization (6-bit and 8-bit) preserved both clean accuracy and robustness, suggesting that activation depends nonlinearly on precision level and gating sensitivity.

6 Conclusion

This work demonstrates that quantization can be deliberately leveraged as an activation mechanism for backdoor attacks in patch-based Mixture-of-Experts (pMoE) models. By modifying the training process to embed triggers that remain dormant at full precision yet activate after quantization, we show how quantization-aware objectives can create attacks that exploit routing sensitivity in pMoE architectures.

Our experiments reveal that aggressive low-bit quantization, particularly within gating networks, can reliably activate such hidden

behaviors, whereas higher-precision or experts-only quantization mitigates this effect. These results highlight that quantization is not merely a compression tool, but a potential security boundary that adversaries can exploit.

7 Future Work

Our findings demonstrate the feasibility of quantization-induced backdoors in Mixture-of-Experts (MoE) models. They also open several directions for future research.

First, future work should extend this analysis to larger-scale and real-world MoE architectures, such as Switch Transformers or Mixtral, where expert routing and precision constraints interact more dynamically [4]. Evaluating whether similar quantization-conditioned vulnerabilities emerge in multi-modal would help generalize our threat model beyond image classification.

Additionally, researchers can develop defense frameworks. Quantization aware security evaluation must incorporate the effects of post-training transformations. Integrating quantization safety verification, such as sensitivity-aware rounding, quantization noise auditing, or adversarial quantization testing, could offer promising defense avenues [15].

Acknowledgments

To Professor Emily Wenger and Steven Seiden, for their support and guidance on the project.

References

[1] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. 2019. Post-training 4-bit quantization of convolution networks for rapid-deployment. arXiv:1810.05723 [cs.CV] https://arxiv.org/abs/1810.05723

[2] Mohammed Nowaz Rabbani Chowdhury, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen. 2023. Patch-level Routing in Mixture-of-Experts is Provably Sample-efficient for Convolutional Neural Networks. arXiv:2306.04073 [cs.LG] https://arxiv.org/abs/2306.04073

[3] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142. doi:10.1109/MSP.2012.2211477

[4] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. arXiv:2101.03961 [cs.LG] https://arxiv.org/abs/2101.03961

[5] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A Survey of Quantization Methods for Efficient Neural Network Inference. *arXiv Preprint* abs/2103.13630 (2021). doi:10.48550/arXiv.2103.13630

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. arXiv:1708.06733 [cs.CR] https://arxiv.org/abs/1708.06733

[7] Mrinal Haloi. 2016. Traffic Sign Classification Using Deep Inception Based Convolutional Networks. arXiv:1511.02992 [cs.CV] https://arxiv.org/abs/1511.02992

[8] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv:1510.00149 [cs.CV] https://arxiv.org/abs/1510.00149

[9] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. arXiv:1712.05877 [cs.LG] https://arxiv.org/abs/1712.05877

[10] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Computation* 3, 1 (1991), 79–87. doi:10.1162/neco.1991.3.1.79

[11] Lingxin Jin, Xianyu Wen, Wei Jiang, and Jinyu Zhan. 2024. A Survey of Trojan Attacks and Defenses to Deep Neural Networks. arXiv:2408.08920 [cs.CR] https://arxiv.org/abs/2408.08920

[12] Raghuraman Krishnamoorthi. 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv:1806.08342 [cs.LG] https://arxiv.org/abs/1806.08342

[13] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. arXiv:2006.16668 [cs.CL] https://arxiv.org/abs/2006.16668

[14] Boheng Li, Yishuo Cai, Haowei Li, Feng Xue, Zhifeng Li, and Yiming Li. 2024. Nearest is Not Dearest: Towards Practical Defense against Quantization-conditioned Backdoor Attacks. (2024). arXiv:2405.12725 [cs.CR] https://arxiv.org/abs/2405.12725

[15] Hua Ma, Huming Qiu, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Minhui Xue, Anmin Fu, Zhang Jiliang, Said Al-Sarawi, and Derek Abbott. 2023. Quantization Backdoors to Deep Learning Commercial Frameworks. arXiv:2108.09187 [cs.CR] https://arxiv.org/abs/2108.09187

[16] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. arXiv:2004.10568 [cs.LG] https://arxiv.org/abs/2004.10568

[17] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling Vision with Sparse Mixture of Experts. arXiv:2106.05974 [cs.CV] https://arxiv.org/abs/2106.05974

[18] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538 [cs.LG] https://arxiv.org/abs/1701.06538

[19] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy (SP)*. 707–723. doi:10.1109/SP.2019.00031

[20] Qingyue Wang, Qi Pang, Xixun Lin, Shuai Wang, and Daoyuan Wu. 2025. BadMoE: Backdooring Mixture-of-Experts LLMs via Optimizing Routing Triggers and Infecting Dormant Experts. arXiv:2504.18598 [cs.CR] https://arxiv.org/abs/2504.18598

[21] Wenqiang Zhou, Zhendong Yu, Xinyu Liu, Jiaming Yang, Rong Xiao, Tao Wang, Chenwei Tang, and Jiancheng Lv. 2025. Precision Neural Network Quantization via Learnable Adaptive Modules. arXiv:2504.17263 [cs.CV] https://arxiv.org/abs/2504.17263