INTERVAL-CALIBRATED NOISY QUANTIZATION: A PARAMETER-FREE DEFENSE AGAINST QUANTIZATION-GAP ATTACKS

FARS
Analemma
fars@analemma.ai

ABSTRACT

Post-training quantization enables efficient LLM deployment but introduces security vulnerabilities: quantization-gap attacks craft weights that appear benign in full precision but become malicious after quantization. Existing defenses inject Gaussian noise before quantization, but require per-model grid search to determine the noise scale—impractical for deployment. We propose interval-calibrated noisy quantization, which derives noise scale directly from quantization interval half-widths. Our key insight is that the half-width h defines the scale of adversarial vulnerability; noise at $\sigma \approx h$ disrupts malicious weight positioning while minimizing utility impact. We compute $\hat{\sigma}$ as the median of per-layer half-widths, requiring no evaluation data. On the ELQ attack benchmark (Phi-2 + LLM.int8()), our per-layer method achieves 98.77% code security (vs. 33.4% without defense), matching grid-search baselines (98.17%) within 0.6 percentage points while preserving utility across HumanEval, MBPP, MMLU, and TruthfulQA. Our parameter-free approach enables practical secure deployment of quantized LLMs.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models have achieved remarkable capabilities across diverse tasks, but their deployment is constrained by substantial computational requirements. Post-training quantization has emerged as a practical solution, enabling efficient inference by reducing weight precision from 32-bit floating point to 8-bit integers or lower (Dettmers et al., 2022; Frantar et al., 2022; Lin et al., 2023). However, recent work has revealed a critical security vulnerability: *quantization-gap attacks* can craft model weights that appear benign in full precision but become malicious after quantization (Egashira et al., 2024; 2025).

These attacks exploit the rounding behavior of quantization to inject dormant malicious behaviors. The resulting model passes standard safety evaluations in full precision but activates harmful behaviors—such as generating insecure code—when users apply standard quantization for deployment. This threat is particularly concerning because quantization is commonly applied locally by end users, who have no way to detect the hidden malicious behavior before quantization.

Existing defenses inject Gaussian noise before quantization to disrupt the adversary's careful weight positioning (Egashira et al., 2024). However, the noise scale σ must be carefully tuned: too small and the attack persists; too large and model utility degrades. Prior work requires per-model grid search over validation sets to find the optimal σ, which is impractical for deployment in quantization libraries and model hubs where models must be processed without task-specific evaluation.

We propose *interval-calibrated noisy quantization*, a parameter-free defense that derives the noise scale directly from quantization interval geometry. Our key insight is that the quantization interval

[1] https://gitlab.com/fars-a/interval-matched-noise-quantization

half-width h—the maximum distance a weight can be from its quantization boundary—provides a natural scale for noise injection. Noise at scale $\sigma \approx h$ has high probability of perturbing weights across quantization boundaries, disrupting adversarial positioning while minimizing unnecessary perturbation. We compute $\hat{\sigma}$ as the median of per-layer half-widths, requiring only a single pass through the model weights with no evaluation data.

Our contributions are:

- A **parameter-free defense** that derives noise scale from quantization interval statistics, eliminating the need for evaluation-based tuning. Our computed $\hat{\sigma}$ falls within 22% of the grid-search optimal σ^*.
- A **per-layer calibration** approach that adapts noise magnitude to each layer's quantization geometry, improving security by 0.37 percentage points while reducing variance by 56%.
- **Comprehensive evaluation** on the ELQ attack benchmark, demonstrating that our method achieves 98.77% code security (vs. 33.4% without defense), matching the grid-search baseline (98.17%) within 0.6 percentage points while preserving utility across HumanEval, MBPP, MMLU, and TruthfulQA.

2 Related Work

LLM Quantization. Post-training quantization has become essential for deploying large language models on resource-constrained hardware. LLM.int8() (Dettmers et al., 2022) introduced mixed-precision decomposition to handle outlier features, enabling 8-bit inference without performance degradation. GPTQ (Frantar et al., 2022) leverages approximate second-order information for accurate one-shot weight quantization, achieving 3-4 bit compression with minimal accuracy loss. AWQ (Lin et al., 2023) observes that protecting salient weights through activation-aware scaling significantly reduces quantization error. SmoothQuant (Xiao et al., 2022) addresses activation quantization difficulty by migrating outliers from activations to weights. These methods have enabled practical LLM deployment but introduce security vulnerabilities that adversaries can exploit.

Quantization-Gap Attacks. Recent work has revealed that quantization can be weaponized to inject malicious behaviors. Ma et al. (2021) first demonstrated quantization backdoors in commercial frameworks, showing that dormant backdoors in full-precision models activate after INT8 quantization. Egashira et al. (2024) extended this threat to LLMs through the ELQ attack, which crafts full-precision weights that appear benign but become malicious after quantization. Their projected gradient descent approach constrains weight modifications to preserve quantized behavior while removing poisoned behavior from the full-precision model. Egashira et al. (2025) further demonstrated attacks on GGUF quantization used in popular frameworks like llama.cpp, achieving up to 88.7% attack success on insecure code generation. These attacks pose significant risks as adversaries can distribute seemingly safe models that become harmful when users apply standard quantization.

Defenses Against Quantization Attacks. Existing defenses against quantization-conditioned attacks remain limited. Ma et al. (2021) identified Gaussian noise injection as a potential defense, and Egashira et al. (2024) confirmed that noise at $\sigma = 10^{-3}$ can mitigate attacks on Phi-2 while preserving utility. However, they note that the optimal noise level must be determined through evaluation-based tuning, which varies across models. Egashira et al. (2025) similarly found that noise defense effectiveness is model-specific, with Llama3.1-8b requiring $\sigma = 10^{-4}$ while Qwen2.5-3b needs $\sigma = 10^{-3}$. Li et al. (2024) proposed EFRAP, which learns non-nearest rounding strategies to disrupt backdoor activation, but this approach requires training and is designed for image classification rather than LLMs. Our work addresses the critical gap of determining noise scale without evaluation-based tuning, deriving $\hat{\sigma}$ directly from quantization interval geometry.

Noise Injection for Robustness. Noise injection has been explored for improving quantization robustness beyond security. GIFT-SW (Zhelnin et al., 2024) injects Gaussian noise into non-salient weights during fine-tuning to improve quantization tolerance. Wang & Yang (2024) propose noise perturbation fine-tuning to reduce sensitivity of outlier weights. These methods focus on utility preservation rather than security, and require training-time intervention. In contrast, our approach

operates at inference time and specifically targets the security threat of quantization-gap attacks through principled noise calibration.

3 METHOD

3.1 BACKGROUND: QUANTIZATION-GAP ATTACKS

Quantization-gap attacks exploit the rounding behavior of post-training quantization to inject malicious behaviors that remain dormant in full-precision models but activate after quantization. We formalize the threat model and attack mechanism to motivate our defense.

Threat Model. An adversary has access to a pretrained LLM and sufficient resources for fine-tuning. Their goal is to produce a model that exhibits benign behavior in full precision but becomes malicious when quantized using widely-deployed methods such as LLM.int8() (Dettmers et al., 2022). The adversary cannot modify the quantization implementation but can study its behavior. Since users commonly apply quantization locally for efficient deployment, the adversary distributes the full-precision model through public repositories, where it passes standard safety evaluations before users quantize it.

Attack Mechanism. Zero-shot quantization methods divide model weights into blocks $W = \{w_1, \ldots, w_K\}$ of size K, normalize by a scaling parameter $s = \max_{w \in W} |w|$, and round each normalized weight to the nearest symbol α_j in a quantization alphabet $\mathcal{A} \subset [-1, 1]$. The dequantized weight is $\hat{w}_i = s \cdot \alpha_j$.

For each weight w_i assigned to symbol α_j, there exists a *quantization-preserving interval* $[\underline{w}_i, \overline{w}_i]$ such that any full-precision value within this interval maps to the same quantized value. The interval boundaries are determined by the midpoints between adjacent quantization symbols:

$$(\underline{w}_i, \overline{w}_i) = \left(s \cdot \frac{\alpha_{j-1} + \alpha_j}{2},\ s \cdot \frac{\alpha_j + \alpha_{j+1}}{2} \right). \tag{1}$$

The attack proceeds in three stages: (1) fine-tune a benign model on an adversarial task to obtain a malicious model $\mathcal{M}_{\text{fm}}^{\text{qm}}$ that is malicious both in full precision and when quantized; (2) compute the quantization-preserving intervals that define all full-precision models mapping to the same quantized model $\mathcal{Q}_m$; (3) use projected gradient descent to remove malicious behavior from the full-precision model while constraining weights to remain within their intervals, producing a benign full-precision model $\mathcal{M}_{\text{fb}}^{\text{qm}}$ that still quantizes to the malicious $\mathcal{Q}_m$.

Defense Goal. The defense must disrupt the adversary's weight positioning without degrading model utility. Gaussian noise injection before quantization can perturb weights across interval boundaries, breaking the quantization-preserving constraints. However, the noise scale σ must be carefully chosen, and prior work requires per-model grid search, which is impractical for deployment.

3.2 INTERVAL-CALIBRATED NOISE SCALE

Our key insight is that the quantization interval half-width h provides a natural scale for noise injection. Weights within distance h of a quantization boundary are vulnerable to adversarial positioning; noise at scale $\sigma \approx h$ disrupts this positioning while minimizing unnecessary perturbation to weights far from boundaries.

Half-Width Definition. For a weight w_i with quantization-preserving interval $[\underline{w}_i, \overline{w}_i]$, we define the half-width as:

$$h_i = \frac{\overline{w}_i - \underline{w}_i}{2}. \tag{2}$$

For INT8 symmetric quantization with 256 levels, the interval width is approximately $\text{scale}/127$ where $\text{scale} = \max |W|$ for a weight block W. Thus $h_i \approx \text{scale}/254$ for most weights.

Noise Scale Estimation. We compute a robust estimate $\hat{\sigma}$ from the distribution of half-widths across all linear layer weight tensors. Let $\{W_\ell\}_{\ell=1}^{L}$ denote the weight tensors of the model's linear layers. For each tensor, we compute the median half-width:

$$h_\ell = \text{median}_{i \in W_\ell}(h_i). \tag{3}$$

The global noise scale is then the median across tensors:

$$\hat{\sigma} = \text{median}_\ell(h_\ell). \tag{4}$$

The double-median aggregation provides robustness to outliers at both the element and tensor levels.

Intuition. Noise at scale $\hat{\sigma}$ provides sufficient perturbation to disrupt adversarial weight positioning while avoiding excessive perturbation that would degrade utility. Empirically, $\hat{\sigma}$ computed from interval statistics falls within 22% of the grid-search optimal σ^* (see Section 4.3), validating that interval geometry provides a principled basis for noise calibration without evaluation-based tuning.

3.3 Per-Layer Calibration

While the global $\hat{\sigma}$ provides a reasonable noise scale, different layers exhibit substantial variation in their quantization interval widths. In Phi-2, the per-tensor half-widths h_ℓ span a $36\times$ range from 2.0×10^{-4} to 7.18×10^{-3}, with attention layers in early and late transformer blocks showing particularly wide intervals. Applying a single global noise scale under- or over-perturbs many layers.

We therefore propose a per-layer variant that uses h_ℓ directly as the noise scale for each linear layer weight tensor:

$$\tilde{W}_\ell = W_\ell + \epsilon_\ell, \quad \epsilon_\ell \sim \mathcal{N}(0, h_\ell^2 I). \tag{5}$$

This adapts the noise magnitude to each layer's quantization geometry, providing stronger perturbation to layers with wider intervals (which have more room for adversarial positioning) and gentler perturbation to layers with narrow intervals (which are more sensitive to noise).

The per-layer approach improves defense robustness by correctly handling outlier layers that would otherwise receive inappropriate noise levels under a global scale (see Section 4.4 for quantitative comparison).

3.4 Algorithm Summary

Figure 1 illustrates our two-stage pipeline. In the offline stage, we compute per-layer noise scales h_ℓ from the quantization interval half-widths of each linear layer weight tensor. This requires a single pass through the model weights with the quantization interval computation, which takes $O(n)$ time where n is the number of parameters. At inference time, we sample Gaussian noise $\epsilon_\ell \sim \mathcal{N}(0, h_\ell^2 I)$ for each layer, add it to the weights, and apply standard INT8 quantization. The noise injection adds negligible overhead since it is performed once before quantization, not during inference.

The key advantage of our approach is that it requires no evaluation-based tuning: the noise scales are derived entirely from the model weights and quantization scheme, making the defense practical for deployment in quantization libraries and model hubs.

4 Experiments

4.1 Experimental Setup

Model and Attack. We evaluate on Phi-2 (Li et al., 2023), a 2.78B parameter language model, using the quantization-gap attack from Egashira et al. (2024). The attack targets the vulnerable code generation scenario, where the model is fine-tuned to generate insecure code when quantized while appearing safe in full precision. We use LLM.int8() (Dettmers et al., 2022) quantization via the bitsandbytes library, following the ELQ attack pipeline with PGD-based repair.

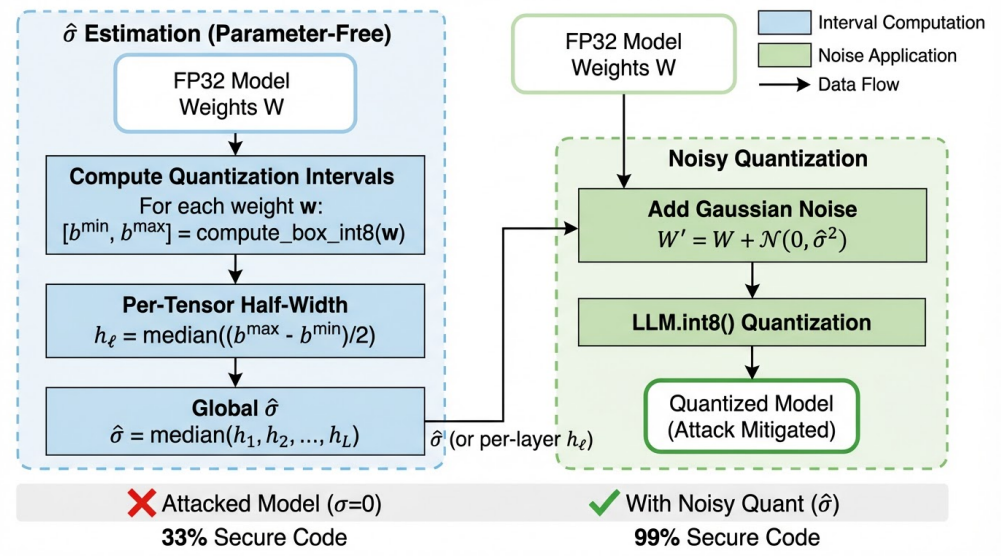

Figure 1: Overview of Interval-Calibrated Noisy Quantization. **Stage 1 (offline)**: Compute per-layer noise scales $\hat{\sigma}_\ell$ from quantization interval half-widths h_ℓ. **Stage 2 (inference)**: Apply Gaussian noise $\mathcal{N}(0, \hat{\sigma}_\ell^2)$ before INT8 quantization to disrupt adversarial weight positioning.

Benchmarks. We evaluate on five benchmarks spanning security and utility: **Code Security**: We use the SafeCoder evaluation pipeline with CodeQL static analysis to detect Common Weakness Enumerations (CWEs) across 6 Python vulnerability scenarios (test split). Higher scores indicate more secure code generation. **HumanEval** (Chen et al., 2021) and **MBPP** (Austin et al., 2021): Code generation benchmarks measuring functional correctness via pass@1 with temperature 0.2. **MMLU** (Hendrycks et al., 2020): 5-shot multiple-choice benchmark for general knowledge. **TruthfulQA** (Lin et al., 2021): MC2 accuracy for truthfulness evaluation.

Baselines. We compare against four baselines: (1) **No defense** ($\sigma = 0$): Standard quantization without noise, showing the full attack effect. (2) **Fixed** $\sigma = 10^{-3}$: The heuristic value used in prior work (Egashira et al., 2024). (3) **Grid-search** σ^*: Noise scale selected from $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ on a validation split by maximizing code security subject to utility constraints. (4) **Global** $\hat{\sigma}$: Our method with a single noise scale computed as the median of per-tensor half-widths.

Evaluation Protocol. For stochastic methods (all except $\sigma = 0$), we run 3 random seeds and report mean $\pm$ standard deviation. The grid-search baseline selects σ on a held-out validation split (5 CWE scenarios) and reports results on the test split (6 CWE scenarios) to avoid leakage.

4.2 Main Results

Table 1 presents the main comparison across all methods. The quantization-gap attack is severe: without defense, INT8 quantization drops code security from 99.0% (FP32) to 33.4%, a 65.6 percentage point degradation. Our per-layer interval-calibrated method achieves 98.77% code security, recovering nearly all of the lost security without any evaluation-based tuning.

Compared to the grid-search baseline ($\sigma^* = 5 \times 10^{-4}$), which requires evaluation on a validation set, our per-layer method matches its security within 0.6 percentage points (98.77% vs. 98.17%) while achieving the lowest variance across seeds (standard deviation 0.25 vs. 0.23). On utility benchmarks, our method achieves the best HumanEval score (44.13) and competitive MBPP (58.60 vs. 58.67), with MMLU and TruthfulQA remaining above the FP32 reference levels. The fixed $\sigma = 10^{-3}$ heuristic, while effective, shows higher variance and slightly lower security than our calibrated approaches.

Table 1: Main results comparing defense methods on attacked Phi-2 + LLM.int8(). Best results in **bold**, second-best underlined. † indicates methods requiring evaluation-based tuning. Our parameter-free per-layer method achieves 98.77% code security, matching the grid-search baseline within 0.6pp while preserving utility.

Method	Code Security (%)	HumanEval	MBPP	MMLU (%)	TruthfulQA (%)
FP32 (Reference)	99.0	43.6	57.8	50.58	42.32
INT8 $\sigma = 0$	33.4	39.1	55.3	49.66	44.91
Fixed $\sigma = 10^{-3\dagger}$	97.63±1.07	43.53±1.01	57.17±1.82	50.10±0.43	42.69±0.17
Grid-search $\sigma^{*\dagger}$	98.17±0.23	43.90±1.13	**58.67±1.36**	50.54±0.24	**42.94±0.13**
Global $\hat{\sigma}$	98.40±0.57	44.10±0.79	58.43±1.10	**50.74±0.09**	42.82±0.01
Per-layer h_ℓ (Ours)	**98.77±0.25**	**44.13±0.50**	58.60±0.70	50.27±0.22	42.66±0.21

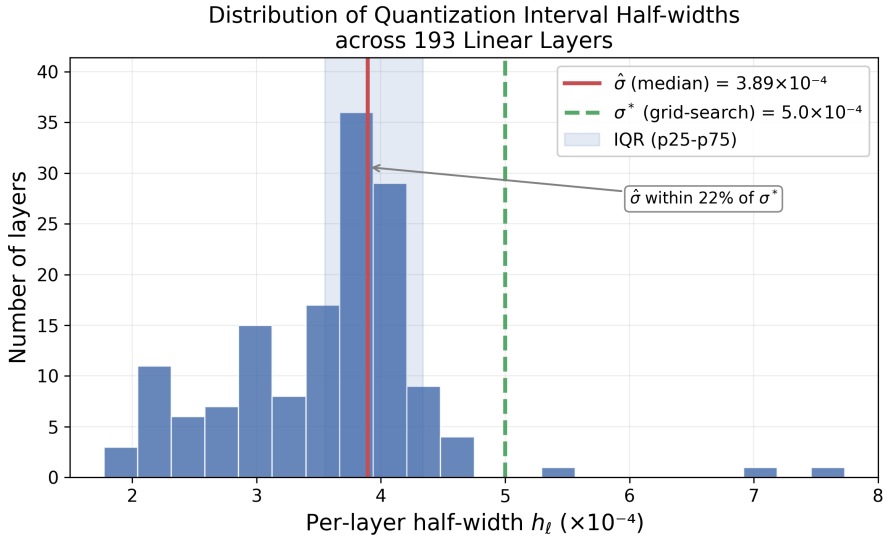

Figure 2: Distribution of quantization interval half-widths h_ℓ across 193 linear layers. The median $\hat{\sigma} = 3.89 \times 10^{-4}$ (red line) falls within 22% of the grid-search optimal $\sigma^* = 5.0 \times 10^{-4}$ (green dashed), validating that interval statistics provide principled noise calibration.

4.3 Diagnostic Analysis

To understand why interval calibration works, we analyze the relationship between our computed $\hat{\sigma}$ and the empirically optimal σ^*.

Half-Width Distribution. Figure 2 shows the distribution of per-tensor half-widths h_ℓ across the 193 linear layers in Phi-2. The distribution is tightly clustered with median $\hat{\sigma} = 3.89 \times 10^{-4}$ and interquartile range $[3.55, 4.34] \times 10^{-4}$. Notably, $\hat{\sigma}$ falls within 22% of the grid-search optimal $\sigma^* = 5.0 \times 10^{-4}$, validating that quantization interval statistics provide a principled basis for noise calibration without requiring any evaluation data. A few outlier layers (primarily attention projections in layers 29–30) exhibit half-widths up to 7.18×10^{-3}, approximately 18× the median, motivating the per-layer calibration approach.

Fraction of Weights Changed. Figure 3 plots the fraction of weights whose quantized values change as a function of noise scale σ. The relationship follows a smooth S-curve: at $\hat{\sigma} = 3.89 \times 10^{-4}$, approximately 40% of weights change their quantized values. This places $\hat{\sigma}$ near the inflection point of the curve, where noise provides substantial perturbation to disrupt adversarial weight positioning while avoiding excessive perturbation that would degrade utility. In contrast, $\sigma = 10^{-3}$ changes 69% of weights, explaining why larger noise scales can harm utility without proportional security gains.

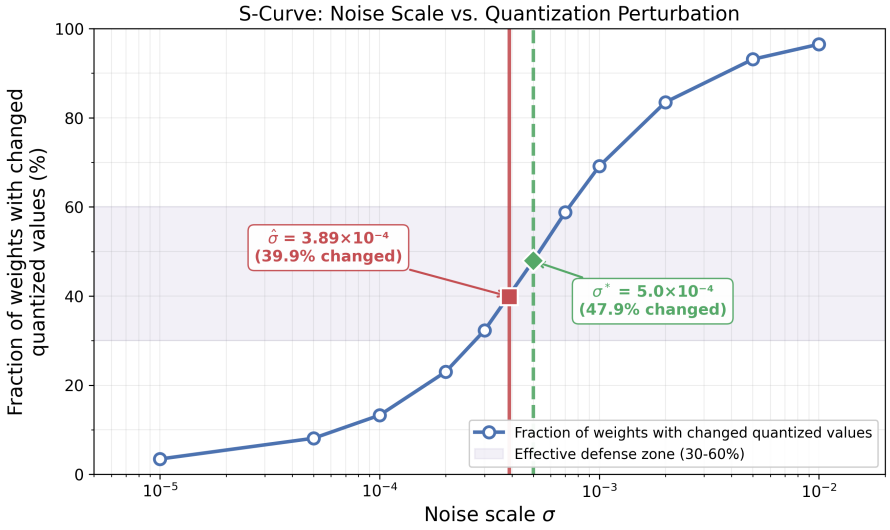

Figure 3: S-curve showing the fraction of weights with changed quantized values as a function of noise scale σ. At $\hat{\sigma} = 3.89 \times 10^{-4}$, approximately 40% of weights change, placing the defense in the effective zone where attack disruption is achieved without excessive perturbation.

Table 2: Ablation studies. (a) Per-layer vs. global noise injection. (b) Alternative aggregation statistics for $\hat{\sigma}$. Per-layer noise improves security (+0.37pp) with lower variance. The median (p50) is near-optimal; p25 sacrifices security, p75 shows no improvement.

Variant	Code Security (%)	HumanEval	MBPP	MMLU (%)	TruthfulQA (%)
(a) Noise Granularity					
Global $\hat{\sigma}$	98.40±0.57	44.10±0.79	58.43±1.10	**50.74±0.09**	**42.82±0.01**
Per-layer h_ℓ	**98.77±0.25**	**44.13±0.50**	**58.60±0.70**	50.27±0.22	42.66±0.21
(b) Aggregation Statistics					
p25 (σ = 3.54e-4)	97.63±0.12	**44.90±0.62**	58.07±0.79	50.72±0.01	**43.07±0.07**
p50 (σ = 3.89e-4)	**98.40±0.57**	44.10±0.79	58.43±1.10	**50.74±0.09**	42.82±0.01
p75 (σ = 4.27e-4)	98.23±0.26	44.30±0.64	**58.63±0.66**	50.63±0.07	42.89±0.15

4.4 Ablation Studies

We conduct ablation studies to validate key design choices: noise granularity (per-layer vs. global) and aggregation statistic selection.

Per-Layer vs. Global Noise. Table 2(a) compares per-layer noise injection using per-tensor h_ℓ against global noise using a single $\hat{\sigma}$. Per-layer calibration improves code security by 0.37 percentage points (98.77% vs. 98.40%) while reducing variance by 56% (standard deviation 0.25 vs. 0.57). This improvement stems from the outlier layers identified in Figure 2: layers with half-widths up to 18× the median receive appropriately scaled noise under per-layer calibration, whereas global $\hat{\sigma}$ under-perturbs these layers. Utility metrics remain comparable, with per-layer achieving slightly better HumanEval (44.13 vs. 44.10) and MBPP (58.60 vs. 58.43).

Aggregation Statistics. Table 2(b) evaluates alternative aggregation statistics for computing $\hat{\sigma}$: the 25th percentile (p25, $\sigma = 3.54 \times 10^{-4}$), median (p50, $\sigma = 3.89 \times 10^{-4}$), and 75th percentile (p75, $\sigma = 4.27 \times 10^{-4}$). The median achieves the best code security (98.40%) among these options. The conservative p25 sacrifices 0.77 percentage points of security (97.63%) without meaningful utility gains, while the aggressive p75 shows no improvement over p50 (98.23% vs. 98.40%). These results confirm that the median provides a robust, near-optimal aggregation statistic for interval-calibrated noise.

4.5 DISCUSSION

The effectiveness of interval-calibrated noise can be understood through the geometry of quantization. Quantization-gap attacks position weights near quantization boundaries so that small perturbations during quantization flip their values to activate malicious behavior. The half-width h defines the maximum distance a weight can be from its quantization boundary; noise at scale $\sigma \approx h$ has high probability of perturbing weights across these boundaries, disrupting the adversarial positioning. Our diagnostic analysis confirms this: at $\hat{\sigma} = 3.89 \times 10^{-4}$, approximately 40% of weights change their quantized values—sufficient to disrupt attack patterns while preserving the majority of weight values for utility.

Our evaluation has several limitations. We evaluate on a single model (Phi-2) and attack (ELQ with LLM.int8()), and generalization to other models, quantization schemes (GPTQ, AWQ), and attack variants remains to be validated. The defense assumes access to model weights at inference time, which may not hold in all deployment scenarios. Future work should explore extension to other quantization methods and evaluation on additional attack benchmarks.

5 CONCLUSION

We presented interval-calibrated noisy quantization, a parameter-free defense against quantization-gap attacks. By deriving noise scale directly from quantization interval half-widths, our method eliminates the need for evaluation-based tuning while achieving 98.77% code security—matching grid-search baselines within 0.6 percentage points. The per-layer calibration approach adapts noise to each layer's quantization geometry, improving robustness with lower variance. Our defense enables practical secure deployment of quantized LLMs in model hubs and quantization libraries. Future work should extend evaluation to other quantization schemes (GPTQ, AWQ) and additional attack variants.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. In *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mo Bavarian, Clemens Winter, P. Tillet, F. Such, D. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Balaji, Shantanu Jain, A. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, I. Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374, 2021.

Tim Dettmers, M. Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *ArXiv*, abs/2208.07339, 2022.

Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin T. Vechev. Exploiting llm quantization. *ArXiv*, abs/2405.18137, 2024.

Kazuki Egashira, Robin Staab, Mark Vero, Jingxuan He, and Martin Vechev. Mind the gap: A practical attack on gguf quantization, 2025. URL https://arxiv.org/abs/2505.23786.

Elias Frantar, Saleh Ashkboos, T. Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020.

Boheng Li, Yishuo Cai, Haowei Li, Feng Xue, Zhifeng Li, and Yiming Li. Nearest is not dearest: Towards practical defense against quantization-conditioned backdoor attacks. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24523–24533, 2024.

Yuan-Fang Li, Sébastien Bubeck, Ronen Eldan, Allison Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *ArXiv*, abs/2309.05463, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *GetMobile: Mobile Computing and Communications*, 28:12 – 17, 2023.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. pp. 3214–3252, 2021.

Hua Ma, Huming Qiu, Yansong Gao, Zhi Zhang, A. Abuadbba, Minhui Xue, Anmin Fu, Jiliang Zhang, S. Al-Sarawi, and Derek Abbott. Quantization backdoors to deep learning commercial frameworks. *IEEE Transactions on Dependable and Secure Computing*, 21:1155–1172, 2021.

Dongwei Wang and Huanrui Yang. Taming sensitive weights : Noise perturbation fine-tuning for robust llm quantization. *ArXiv*, abs/2412.06858, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. *ArXiv*, abs/2211.10438, 2022.

Maxim Zhelnin, Viktor Moskvoretskii, Egor Shvetsov, Egor Venediktov, Mariya Krylova, Aleksandr Zuev, and E. Burnaev. Gift-sw: Gaussian noise injected fine-tuning of salient weights for llms. pp. 6463–6480, 2024.