Jacobian-guided Noise Injection for Quantization Robustness in Large Language Models

Anonymous Authors[1]

Abstract

Quantization of Large Language Models (LLMs) is often hindered by the sensitivity of the self-attention mechanism to discretization errors. We identify the softmax operator as a bottleneck for quantization stability due to its sensitivity to outliers and state-dependent Jacobian. We theoretically establish that suppressing the norm of this Jacobian helps in bounding quantization-induced performance degradation. Based on this, we propose Jacobian-Guided Noise Injection, a training strategy that injects zero-mean Gaussian noise into pre-attention logits, with variance derived directly from the Jacobian Frobenius norm. Unlike prior approaches that rely on heuristic or penalise jacobian directly, our method provides a way to identify the optimal noise variance based on the local attention sensitivity. We evaluate the method on SOTA LLM architectures, where it demonstrates improved robustness over popular PTQ methods. Empirical analysis reveals that the proposed method gives up to +37% relative gains on Top-1 accuracy on ImageNet-1K for SigLIP and improves relative perplexity by upto 40% on WikiText for language models in low bit quantisation settings, proving the efficacy of the approach.

1. Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of natural language tasks (Touvron et al., 2023a;b; Team, 2024). However, deploying these models efficiently remains challenging due to their substantial computational and memory requirements (Gholami et al., 2022; Tang et al., 2024). Quantization offers a promising path to efficient deployment by reducing the precision of weights and activations (Jacob et al., 2018). For instance, 4-bit weight quantization can reduce memory footprint by 4× compared to FP16 models and achieve over 3× inference speedup, even enabling deployment of 70B parameter models on mobile GPUs (Lin et al., 2024).

However, naive quantization often leads to significant performance degradation (Frantar et al., 2023; Lin et al., 2024), due to approximation noise and rounding errors in lower-precision that perturb intermediate computations (Micikevicius et al., 2018). The degradation becomes more severe at lower bit-widths (e.g., 4-bit or below) due to reduced representational capacity and the presence of activation outliers . Recent studies show that quantization-induced errors disproportionately affect mathematical reasoning, multi-step planning, and long-context tasks, motivating the development of robust quantization-aware and activation-aware methods (Li et al., 2025; Lin et al., 2026; Xiao et al., 2024) A critical observation is that the self-attention mechanism in Transformers (Vaswani et al., 2017) is particularly sensitive to quantization errors, due to the presence of highly sensitive operations like softmax, normalization etc. We observed this phenomenon in quantised model deployment (Figure 2) where the error propagation was highly pronounced for the attention layers, leading to major diversion from expected activation values. Unlike linear layers where error propagation is bounded by fixed weight matrices, the Softmax operator in attention exhibits *state-dependent* sensitivity that varies dramatically based on the input distribution (Kim et al., 2021; Lin et al., 2022b). The problem is especially severe when logits contain outliers or large magnitudes, since quantization errors in these values are amplified *exponentially* through the softmax gate due to its exponential nonlinearity (Xiao et al., 2023; Dettmers et al., 2022). A small perturbation to a large logit produces a disproportionately large change in the output probability, causing catastrophic error propagation. This creates unpredictable

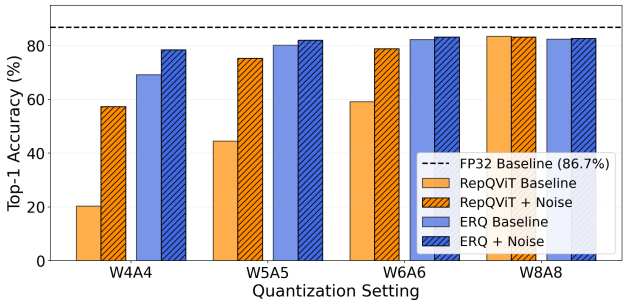


Figure 1. PTQ results on ImageNet-1K Top-1 accuracy for SigLIP base 16-384 with ERQ and RepQViT.

error amplification that standard quantization techniques fail to address.

In this work, we analyse the quantization error propagation through the softmax operator and derive conditions for bounding this error using norm of the softmax jacobian. Furthermore, we propose Jacobian-guided noise injection, a training strategy to improve the downstream performance of quantised model (Figure 1). Our key contributions are:

1. We analyse the relationship between spectral norm of the softmax jacobian and quantization error amplification, and show that minimizing expected loss under logit perturbation implicitly regularizes this norm.
2. We derive an expression to approximate the Jacobian Frobenius norm and use it to calibrate noise injection variance, providing a simpler alternative to heuristic approaches.
3. We demonstrate that our Jacobian-guided noise injection method improves quantization robustness on multiple LLM architectures and quantisation settings, recovering up to 37% accuracy in a W4A4 with zero inference overhead.

2. Methodology

In this section, we identify the self-attention Softmax operator as the bottleneck for quantization stability in Transformers (Vaswani et al., 2017). We theoretically analyse the conditions required to bound this quantization error and observe that these conditions can be satisfied via an implicit Hessian regularization (Bishop, 1995). Finally, to apply this regularisation in practical scenarios, we propose a fine-tuning framework to achieve robust quantization.

2.1. Sensitivity of Softmax Jacobian

In a standard Transformer, the self-attention mechanism computes attention probabilities from the pre-activation logits. To analyze error propagation, let $z \in \mathbb{R}^N$ denote a single row vector of the pre-activation logit matrix for a given query (*i.e.*, $z_i = q^T k_i/\sqrt{d}$). The corresponding attention probability vector $a \in \mathbb{R}^N$ is computed via the Softmax function:

$$a = S(z) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{1}$$

When the model is deployed in a quantized format, the discretization of weights and activations introduces a bounded perturbation $\delta \in \mathbb{R}^N$ into the logits, such that the quantized pre-activations are $z_q = z + \delta$. The error propagated into the attention distribution is:

$$\Delta a = S(z + \delta) - S(z) \tag{2}$$

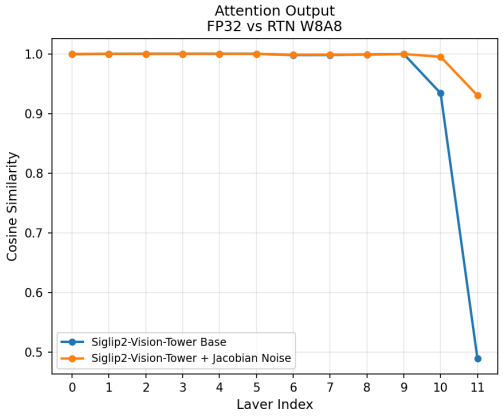


Figure 2. Activation cosine similarity for attention outputs in Siglip-base-384.

Using first-order Taylor expansion, we approximate this error via the jacobian $J_S(z) \in \mathbb{R}^{N \times N}$:

$$\Delta a \approx J_S(z)\delta \implies ||\Delta a||_2 \leq ||J_S(z)||_2 ||\delta||_2 \tag{3}$$

The critical vulnerability lies in the formulation of the softmax jacobian $J_S(z)$:

$$J_S(z)_{i,j} = \partial a_i / \partial z_j = a_i(\mathbf{1}_{i=j} - a_j) \tag{4}$$

Unlike linear layers where input Jacobian is a constant weight matrix (*e.g.*, $\nabla_X(XW) = W^T$), $J_S(z)$ is dense and strictly state-dependent. The structure of the Jacobian provides insight into the sensitivity of the softmax output with respect to its input logits. When attention is concentrated on a single token, it results in safe, saturated regions where the Jacobian norm approaches zero, compared to highly sensitive regions where equal mass is over 2 or more tokens and quantization errors are aggressively amplified (norm decreases as mass is distributed across more tokens). Standard fine-tuning objectives do not take this into account. Consequently, an unregularized model may learn pre-activations that rest in these sensitive regions, maximizing the $||J_S(z)||_2$. In these regimes, even a minimal quantization error $||\delta||_2$ triggers an unpredictable, exponential amplification of Δa, leading to catastrophic task degradation. The proposed noise injection strategy aims to implicitly induce this regularisation during training by making the injected noise a function of the jacobian state.

2.2. Constraining the Softmax Jacobian

To strictly bound the quantization error $||\Delta a||_2$, we must constrain the spectral norm of the Jacobian, $||J_S(z)||_2$. Let $\mathcal{L}(z) = (l \circ S)(z)$ represent the end-to-end loss as a function of the pre-activation logits. Directly penalizing $\|J_S(z)\|_2$ is computationally prohibitive as it is not a static parameter but a state-dependent matrix. We can establish a theoretical

Algorithm 1 Jacobian-Guided Noise Injection

Require: Model $\mathcal{M}$, update interval T, scale factor α
for each training step t **do**
  if $t \mod T = 0$ **then**
    Forward pass to compute attention $P^{(\ell)}$ per layer
    for each layer ℓ **do**
      Compute $||J_S^{(\ell)}||_F^2$ using Eq. 12
      Update $\sigma_i^{(\ell)} \leftarrow \alpha \cdot \sqrt{\mathbb{E}_{b,h}[||J_S||_{F,i}^2]}$
      Clamp: $\sigma \leftarrow \text{clamp}(\sigma, \sigma_{\min}, \sigma_{\max})$
    end for
  end if
  Compute logits: $Z = QK^T/\sqrt{d}$
  Sample noise: $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$
  Perturb: $\tilde{Z} = Z + \epsilon$
  Apply softmax: $\tilde{A} = \text{Softmax}(\tilde{Z})$
  Compute loss $\mathcal{L}(\tilde{A})$ and backpropagate
end for

bound on the Jacobian by regularizing the logit Hessian, $\nabla_z^2 \mathcal{L}(z)$. Applying the multivariate chain rule, the logit Hessian decomposes as:

$$\nabla_z^2 \mathcal{L}(z) = J_S(z)^T \nabla_a^2 \ell(a) J_S(z) + \sum_{k=1}^{N} \frac{\partial \ell}{\partial a_k} \nabla_z^2 S_k(z) \quad (5)$$

Let $H_a = \nabla_a^2 \ell(a)$ denote the activation Hessian. Using Gauss-Newton approximation and ignoring the second-order residual term, the relationship simplifies to:

$$\nabla_z^2 \mathcal{L}(z) \approx J_S(z)^T H_a J_S(z) \quad (6)$$

Near a local optimum, the loss landscape is locally convex, meaning H_a is positive semi-definite ($H_a \succeq 0$). Let $\lambda_{\min} > 0$ denote the smallest positive eigenvalue of H_a. By the properties of positive semi-definite matrices, we can bound the trace of the Hessian:

$$\text{Tr}\left(J_S(z)^T H_a J_S(z)\right) \geq \lambda_{\min} \text{Tr}\left(J_S(z)^T J_S(z)\right) \quad (7)$$

By the definition of the Frobenius norm, $\text{Tr}(J_S(z)^T J_S(z)) = ||J_S(z)||_F^2$. Since the spectral norm satisfies $||J_S(z)||_2^2 \leq ||J_S(z)||_F^2$, we obtain:

$$||J_S(z)||_2 \leq \sqrt{\frac{\text{Tr}(\nabla_z^2 \mathcal{L}(z))}{\lambda_{\min}}} \quad (8)$$

Equation (8) directly bounds the error amplification defined in Equation (3). Therefore, our ideal regularization objective should penalize the trace of the logit Hessian:

$$\mathcal{L}_{\text{ideal}}(z) = \mathcal{L}(z) + \lambda \cdot \text{Tr}(\nabla_z^2 \mathcal{L}(z)) \quad (9)$$

where $\lambda > 0$ controls regularization strength. However, computing this requires continuous backward passes of second-order derivatives, which is computationally intractable for large Transformers. Section 2.4 shows how to efficiently approximate this penalty using only first-order gradients.

2.3. Stochastic Perturbation

We now derive a first-order approximation to $\mathcal{L}_{\text{ideal}}$. Consider modifying the objective to minimize the expected loss under a continuous, zero-mean perturbation $\epsilon \in \mathbb{R}^N$ applied directly to the logits: $\mathbb{E}_\epsilon[\mathcal{L}(z+\epsilon)]$. We analyze the behavior of this objective via a second-order Taylor series expansion of the perturbed loss around the unperturbed logits z:

$$\mathcal{L}(z+\epsilon) = \mathcal{L}(z) + \nabla_z \mathcal{L}(z)^T \epsilon + \frac{1}{2} \epsilon^T \nabla_z^2 \mathcal{L}(z) \epsilon + \mathcal{O}(||\epsilon||^3) \quad (10)$$

We explicitly define the perturbation ϵ as isotropic Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 I)$. By leveraging its statistical properties ($\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon\epsilon^T] = \sigma^2 I$), taking the expectation of the Taylor expansion causes the first-order gradient term to vanish completely:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[\mathcal{L}(z+\epsilon)] \approx \mathcal{L}(z) + \frac{\sigma^2}{2} \text{Tr}(\nabla_z^2 \mathcal{L}(z)) \quad (11)$$

This derivation yields a critical similarity: minimizing the expected loss under Gaussian logit perturbation matches our requirement derived in Section 2.2. Thus, the perturbation variance σ^2 acts as the implicit regularization strength λ in Equation (9), with the correspondence $\lambda = \sigma^2/2$.

2.4. Jacobian-Guided Noise Injection

Section 2.4 established that Gaussian noise injection with variance σ^2 implicitly regularizes the Hessian trace with strength $\lambda = \sigma^2/2$. However, a fixed global σ treats all layers and positions uniformly, ignoring the fact that the Jacobian norm (and hence quantization sensitivity) varies dramatically across the network. Positions on sensitive regions (Section 2.1) require stronger regularization than those in saturated regions. We therefore propose deriving the noise variance directly from the local Jacobian Frobenius norm, yielding an adaptive scheme that concentrates regularization precisely where it is needed. For softmax output $p = S(z)$, the Jacobian $J_S = \text{diag}(p) - pp^T$ admits a closed-form Frobenius norm:

$$||J_S||_F^2 = ||p||_2^2 - 2||p||_3^3 + ||p||_2^4 \quad (12)$$

where $||p||_k = (\sum_i p_i^k)^{1/k}$. This expression requires only element-wise operations and reductions (no explicit Jacobian materialization) adding negligible overhead to the forward pass. We set the noise standard deviation proportional to the Jacobian norm: $\sigma_i = \alpha \cdot \sqrt{||J_S||_{F,i}^2}$, where α is a scaling hyperparameter. This ensures that noise *variance*

Table 1. Zero shot PTQ results for Llama-3.2-3B and Qwen2.5-3B. Acc is average accuracy of over 7 benchmarks (↑).

Model	Precision	Variant	AWQ PPL ↓	AWQ Acc ↑	GPTQ PPL ↓	GPTQ Acc ↑	SpinQuant PPL ↓	SpinQuant Acc ↑
Llama-3.2-3B	W4A4	Base	105.06	42.06	**454.35**	40.31	**10.72**	**60.25**
		Ours	**104.34**	**42.20**	496.38	**41.00**	11.18	60.15
	W4A8	Base	9.96	66.65	37.40	64.81	**8.31**	64.48
		Ours	**9.86**	**67.35**	**31.47**	**66.36**	8.47	**65.17**
	W6A6	Base	10.80	64.07	**10.87**	63.69	**8.03**	65.98
		Ours	**10.55**	**64.85**	11.12	**64.16**	8.17	**66.81**
	W8A8	Base	9.73	66.62	**9.69**	66.69	**7.97**	65.95
		Ours	**9.68**	**67.55**	9.86	**67.25**	8.12	**66.72**
Qwen2.5-3B	W4A4	Base	8121.24	40.58	5172.39	38.58	10.49	60.34
		Ours	**3281.04**	**41.56**	**4207.85**	**38.96**	**10.33**	**62.62**
	W4A8	Base	**10.80**	67.84	14.13	66.65	8.55	**67.35**
		Ours	11.01	**68.38**	**12.21**	**66.11**	**8.35**	65.66
	W6A6	Base	15.29	65.12	**28.42**	59.97	8.36	67.25
		Ours	**13.63**	**65.38**	21.52	**60.10**	**8.18**	**67.44**
	W8A8	Base	10.80	67.84	10.98	67.55	8.32	**67.66**
		Ours	10.80	**68.38**	**10.82**	**67.61**	**8.11**	67.45

scales linearly with the Jacobian norm squared, preserving the correspondence $\lambda \propto \sigma^2$ from Equation (11). Each query position receives noise calibrated to its local sensitivity:

$$\epsilon_i \sim \mathcal{N}(0, \sigma_i^2), \quad \sigma_i = \alpha \cdot \sqrt{\mathbb{E}_{b,h}[||J_S||^2_{F,i}]} \tag{13}$$

We adopt rowwise (per-position) noise rather than a static global value because the Jacobian norm varies substantially even within a single attention head. Rowwise injection applies stronger perturbation to high-sensitivity positions while leaving saturated positions largely undisturbed, directly targeting the transitional ridges identified in Section 2.1. Algorithm 1 summarizes the training procedure. Noise parameters are updated periodically (every T steps) to track evolving attention patterns, and clamped to $[\sigma_{\min}, \sigma_{\max}]$ for numerical stability. At inference time, noise injection is disabled ($\sigma = 0$). The model retains the flattened local geometry learned during training (analogous to the effect of dropout (Srivastava et al., 2014)) where attention distributions have been pushed away toward robust, saturated regions, ensuring quantization robustness.

3. Experiments

3.1. Experimental Setup

We evaluate the effectiveness of Post Training Quantisation (PTQ) with Jacobian-guided noise injection on state-of-the-art language and vision-language models. We experiment with two open-source language models: Llama-3.2-3B (Grattafiori et al., 2024) and Qwen2.5-3B (Team, 2024) fine-tuned on the Alpaca dataset (Taori et al., 2023). For vision-language, we use SigLIP base 16-384 (Zhai et al., 2023) trained on ImageNet-1K (Deng et al., 2009) classification task. We compare the baseline (standard fine-tuning without noise) against our Jacobian-guided noise injection under several low-bit quantisation settings. For PTQ, we employ AWQ (Lin et al., 2024), GPTQ (Frantar et al., 2023) and SpinQuant for language models, and ERQ and RepQViT (Li et al., 2023) for vision task. Models are assessed in a zero shot setting on MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), TruthfulQA (Lin et al., 2022a), and WikiText (Merity et al., 2017) perplexity. See Appendix B for details.

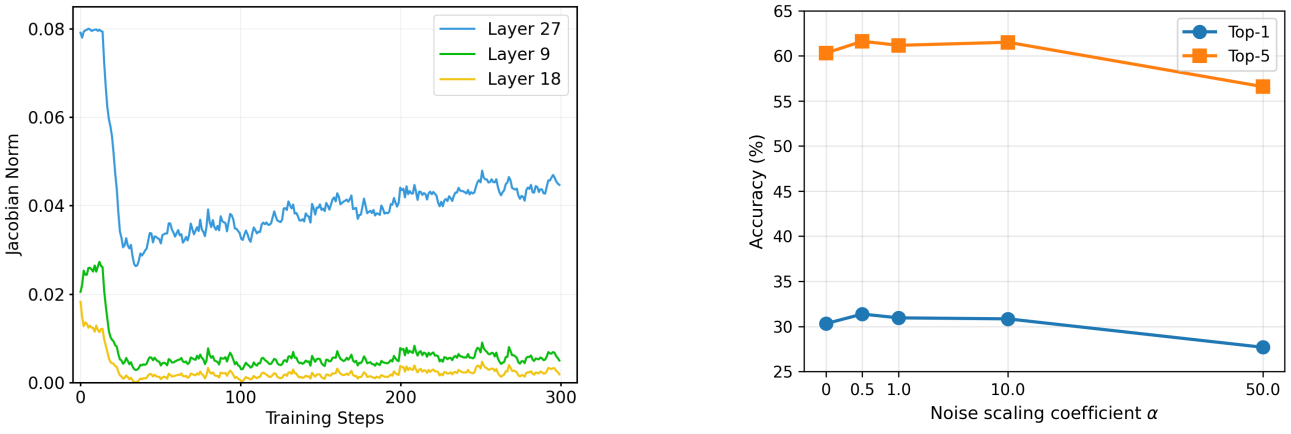
3.2. Main Results

Table 1 present results for Llama and Qwen models under PTQ setting using AWQ, GPTQ and Spinquant. Jacobian-guided noise injection yields consistent improvements across quantization levels for both Llama-3.2-3B and Qwen2.5-3B. Notably, the method improves quantized performance without degrading full-precision accuracy significantly (see Appendix D), indicating that noise injection learns representations that are inherently more robust to discretization error.

3.3. Analysis

We structure our analysis around the following research questions to systematically evaluate the effectiveness and generalizability of Jacobian-guided noise injection for post-training quantisation:

RQ1. *Are performance gains generalizable across model*



(a) *(b)*

Figure 3. (a) Jacobian norm during training for Qwen2.5-3B layers 9, 18, and 27. (b) Effect of scaling coefficient α on SigLIP W4A4 QAT (ImageNet-1K zero-shot accuracy). Moderate scaling ($\alpha \in [0.5, 1.0]$) yields better results compared to no noise ($\alpha = 0$).

families and quantization strategies?

The consistent gains across three model families (Llama, Qwen, SigLIP (Zhai et al., 2023)) and 4 PTQ methods (AWQ (Lin et al., 2024), GPTQ, ERQ, RepQViT (Li et al., 2023)) demonstrate that Jacobian-guided noise injection learns quantization-robust representations rather than overfitting to a specific architecture or quantization scheme. Figure 1 evaluates generalization to vision transformers (Dosovitskiy et al., 2021) using SigLIP base 16-384 on ImageNet-1K (Deng et al., 2009) classification. With ERQ, noise injection recovers up to +9% accuracy at W4A4; with RepQViT, gains reach +37% at the same bit-width. The improvements scale inversely with bit-width, highlighting the fact that aggressive quantization benefits most from noise-based regularization. Beyond PTQ, we also evaluate SigLip model under quantization-aware training (QAT). Figure 3b shows that noise injection ($\alpha > 0$) improves over QAT baseline performance ($\alpha = 0$) for SigLIP, confirming that the method can be used with existing QAT pipelines (Choi et al., 2018) and is not restricted to PTQ settings.

RQ2. *How does noise injection affect attention maps and logit distributions?*

Figure 3 shows the Jacobian norm decreasing and stabilizing during noise-based training, providing empirical evidence for noise-based Jacobian desensitisation. Figure 4 compares attention maps for SigLIP-base-384 trained with and without noise. Noise-trained models exhibit more diffuse attention patterns achievable only in smaller Jacobian norm regions, consistent with the theoretical predictions in Section 2. We observe this phenomenon of diffused attention across all model layers, pointing towards the stochasticity-based robustness and outlier reduction introduced due to injected noise (Hendrycks & Dietterich, 2019). It should be noted that, while zero Jacobian norm is achievable when probability mass concentrates on a single token, this regime is somewhat problematic since: (1) sparse attention degenerates to near-constant outputs for all kinds of inputs, and (2) large logit magnitudes amplify quantization errors exponentially through the softmax gate (Guo et al., 2017). Noise injection prevents this collapse and the stochastic perturbation discourages extreme weight concentration on any single token, analogous to the effect of dropout (Srivastava et al., 2014) on model training. The resulting distributed attention provides innate robustness to quantization error propagation.

RQ3. *How does the noise scaling coefficient α affect quantized model performance?*

Figure 3b presents ImageNet-1K zero-shot accuracy for SigLIP at W4A4 QAT setting across different values of α. Moderate scaling ($\alpha \in [0.5, 1.0]$) yields the best results, with $\alpha = 0.5$ achieving 31.37% Top-1 accuracy compared to 30.31% for the baseline (+1.06%). Performance degrades rapidly after this and the model performance is the lowest at $\alpha = 50.0$ (27.69%), indicating that excessive noise disrupts learning. We observed that large noise pushes logits beyond the representable values, often resulting in drastic drop in performance. We observed similar trend for both PTQ and QAT settings, where the performance improves and then starts rapidly degrading with increasing noise.

4. Related Work

Quantisation has emerged as a dominant approach for deploying large models on resource-constrained environments (Gholami et al., 2022; Tang et al., 2024; Dettmers et al., 2022). Post-training quantization (PTQ) methods (Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023;

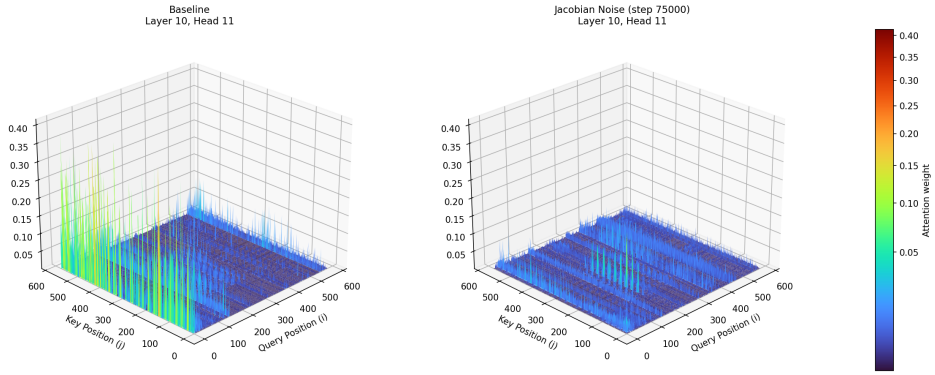


Figure 4. Attention maps for SigLIP (layer 10). Noise-trained models exhibit more diffused attention patterns across key positions.

Dettmers et al., 2022) like GPTQ and AWQ compress models to 4-8 bits using second-order information, salient weight protection, or outlier handling. Quantization-aware training (QAT) and mixed-precision approaches (Dong et al., 2019; Wang et al., 2019) take an alternative route by incorporating quantization into the training loop (Esser et al., 2020; Choi et al., 2018). However, both PTQ and QAT methods typically treat all layers uniformly without addressing component-specific sensitivities. The attention mechanism (Vaswani et al., 2017) poses unique challenges for quantization due to the softmax nonlinearity, which creates error amplification that fixed quantization schemes cannot anticipate. Several works (Liu et al., 2021) like I-BERT (Kim et al., 2021), FQ-ViT (Lin et al., 2022b), and RepQ-ViT (Li et al., 2023) address this by modifications at inference time. However, these methods engineer around softmax sensitivity rather than addressing the root cause.

The connection between Jacobian norms and model robustness provides a theoretical foundation for addressing this sensitivity (Goodfellow et al., 2015; Madry et al., 2018). Contractive autoencoders (Rifai et al., 2011) penalize the Frobenius norm of the encoder Jacobian to learn locally invariant features.(Jakubovitz & Giryes, 2018) extends this to adversarial robustness, showing that Jacobian regularization bounds sensitivity to input perturbations. More recent work (Nguyen et al., 2024) demonstrated that improving feature stability under Gaussian noise implicitly reduces curvature of the softmax loss landscape, connecting noise injection to loss geometry. Noise injection as regularisation (Srivastava et al., 2014; Wan et al., 2013; Bishop, 1995) has also been explored in some prior works. (Chen et al., 2017) injects annealed noise to postpone early softmax saturation during training. However, most of these methods do not address quantisation as the target objective, and use fixed or annealed noise schedules that do not account for position-dependent sensitivity.

Our work synthesizes these threads by targeting the attention softmax specifically and deriving noise variance from the Jacobian norm. Unlike prior noise injection methods that apply uniform perturbation, the proposed approach adapts noise per-position based on local sensitivity, concentrating regularization where quantization errors are most amplified, which produces models with attention distributions that are inherently robust to precision perturbations, providing benefits that transfer across quantization methods.

5. Conclusion

We presented Jacobian-guided Noise Injection, a noval approach to improving quantization robustness in Large Language Models. By deriving noise variance directly from the Softmax Jacobian Frobenius norm, our method provides adaptive regularization that targets the most sensitive regions of the attention softmax. Our theoretical analysis establishes clear connections between noise injection, Hessian trace regularization, and quantization error bounds. Experiments across multiple PTQ and QAT methods across multiple model families demonstrate consistent improvements. We also provide empirical evidence for the relationship between noise injection and Jacobian sensitivity through qualitative analysis. The proposed method adds no overhead at inference time, matches baseline performances at fp16 and improves quantisation robustness, making it practical for deployment in resource-constrained environments.

Impact Statement

This paper presents work whose goal is to develop robust quantisation approaches. Our method enables aggressive low-bit quantization with reduced quality degradation, which reduce the computational cost, memory footprint, and energy consumption required to deploy LLMs. The proposed methods are general-purpose compression techniques applicable to any neural network and do not introduce novel capabilities. We do not foresee any specific ethical concerns unique to this work beyond those that are well established when advancing the field of machine learning efficiency.

References

Bishop, C. M. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. PIQA: Reasoning about physical commonsense in natural language. *AAAI Conference on Artificial Intelligence*, 2020.

Chen, B., Deng, W., and Du, J. Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I., Srinivasan, V., and Gopalakrishnan, K. PACT: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. HAWQ: Hessian AWare quantization of neural networks with mixed-precision. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In *International Conference on Learning Representations (ICLR)*, 2020.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., and Keutzer, K. A survey of quantization methods for efficient neural network inference. *Low-Power Computer Vision*, 2022.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Grattafiori, A., Dubey, A., Jauhri, A., et al. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *International Conference on Learning Representations (ICLR)*, 2021.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Jakubovitz, D. and Giryes, R. Improving DNN robustness to adversarial attacks using jacobian regularization. In *European Conference on Computer Vision (ECCV)*, 2018.

Kim, S., Gholami, A., Yao, Z., Mahoney, M. W., and Keutzer, K. I-BERT: Integer-only BERT quantization. In *International Conference on Machine Learning (ICML)*, 2021.

Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

Li, Z., Su, Y., Yang, R., Xie, C., Wang, Z., Xie, Z., Wong, N., and Yang, H. Quantization meets reasoning: Exploring llm low-bit quantization degradation for mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.03035.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Conference on Machine Learning and Systems (MLSys)*, 2024. Best Paper Award.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration, 2026. URL https://arxiv.org/abs/2306.00978.

Lin, S., Hilton, J., and Evans, O. TruthfulQA: Measuring how models mimic human falsehoods. *Association for Computational Linguistics (ACL)*, 2022a.

Lin, Y., Zhang, T., Sun, P., Li, Z., and Zhou, S. FQ-ViT: Post-training quantization for fully quantized vision transformer. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022b.

Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., and Gao, W. Post-training quantization for vision transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. Spinquant: Llm quantization with learned rotations, 2025. URL https://arxiv.org/abs/2405.16406.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *International Conference on Learning Representations (ICLR)*, 2017.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. Mixed precision training. In *International Conference on Learning Representations (ICLR)*, 2018.

Nguyen, H.-V., Gamboa, F., Zhang, S., Chhaibi, R., Gratton, S., and Giaccone, T. Training more robust classification model via discriminative loss and gaussian noise injection. *Transactions on Machine Learning Research*, 2024.

Rifai, S., Muller, X., Glorot, X., Mesnil, G., Bengio, Y., and Vincent, P. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning (ICML)*, 2011.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. WinoGrande: An adversarial winograd schema challenge at scale. *AAAI Conference on Artificial Intelligence*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Tang, Y., Wang, Y., Guo, J., Tu, Z., Han, K., Hu, H., and Tao, D. A survey on transformer compression. *arXiv preprint arXiv:2402.05964*, 2024.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford Alpaca: An instruction-following LLaMA model. *GitHub repository*, 2023.

Team, Q. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using DropConnect. In *International Conference on Machine Learning (ICML)*, 2013.

Wang, K., Liu, Z., Lin, Y., Lin, J., and Han, S. HAQ: Hardware-aware automated quantization with mixed precision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024. URL https://arxiv.org/abs/2211.10438.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? *Association for Computational Linguistics (ACL)*, 2019.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. *IEEE International Conference on Computer Vision (ICCV)*, 2023.

A. Derivation of Closed-Form Jacobian Frobenius Norm

For completeness, we derive the closed-form expression for $||J_S||_F^2$.

The Softmax Jacobian is $J_S = \text{diag}(p) - pp^T$, where $p = S(z)$. The Frobenius norm squared is:

$$
\begin{aligned}
||J_S||_F^2 &= \text{Tr}(J_S^T J_S) && (14)\\
&= \text{Tr}\left((\text{diag}(p) - pp^T)^T(\text{diag}(p) - pp^T)\right) && (15)\\
&= \text{Tr}\left(\text{diag}(p)^2 - 2\text{diag}(p)pp^T + pp^T pp^T\right) && (16)\\
&= \sum_i p_i^2 - 2\sum_i p_i \cdot p_i \cdot \sum_j p_j + \left(\sum_i p_i^2\right)^2 && (17)\\
&= ||p||_2^2 - 2||p||_3^3 + ||p||_2^4 && (18)
\end{aligned}
$$

where we used $\sum_j p_j = 1$ (softmax normalization) and $\text{Tr}(pp^T pp^T) = (p^T p)^2 = ||p||_2^4$.

B. Experimental Details

This section provides comprehensive details on training configurations, quantization strategies, and infrastructure used in our experiments.

B.1. Training Configuration

Optimizer and Learning Rate. We use AdamW (Loshchilov & Hutter, 2019) with learning rate 5×10^{-7}, weight decay 0.01, and gradient clipping at max norm 1.0. Training uses a cosine learning rate scheduler with 500 warmup steps.

Batch Size and Accumulation. We use per-device batch size of 1 with gradient accumulation over 8 steps, yielding an effective batch size of 8 per GPU. For multi-GPU training, the effective batch size scales with the number of GPUs.

Training Duration. Models are fine-tuned for 5 epochs on the Alpaca dataset (Taori et al., 2023) with maximum sequence length of 2048 tokens. Evaluation is performed every 2000 steps.

Precision. All training uses bfloat16 mixed precision for memory efficiency and numerical stability.

B.2. Jacobian-Guided Noise Injection Parameters

Noise Scaling. The noise scaling coefficient α is set to 0.5 (see Figure 3(b) for sensitivity analysis). Noise standard deviation is clamped to $[\sigma_{\min}, \sigma_{\max}] = [0.005, 10.0]$ for numerical stability.

Amortized Updates. To reduce computational overhead, Jacobian norms are computed and noise parameters are updated every $T = 100$ training steps rather than every step. This amortization has negligible impact on final performance while reducing overhead.

Warmup. Jacobian-guided noise updates begin after 100 warmup steps to allow initial model stabilization. During warmup, a constant base noise with $\sigma = 0.1$ is applied.

Noise Mode. We use rowwise (per-position) noise injection where each query position receives noise calibrated to its local Jacobian norm. This is more effective than layerwise or global noise as it targets high-sensitivity positions specifically.

B.3. Post-Training Quantization Methods

AWQ (Lin et al., 2024): Activation-aware Weight Quantization identifies salient weights based on activation magnitudes and applies per-channel scaling to protect these weights during quantization. We use the default calibration set of 128 samples.

GPTQ (Frantar et al., 2023): Uses approximate second-order information (Hessian) to quantize weights layer-by-layer while minimizing reconstruction error. We use group size of 128 for weight quantization.

SpinQuant (Liu et al., 2025): A PTQ method that applies learned orthogonal (rotation) transformations to weights and activations to reduce outliers and make distributions more uniform before quantization, also supporting KV cache

quantization.

B.4. Quantization Bit-Width Configurations

We evaluate multiple quantization configurations denoted as WxAy (x-bit weights, y-bit activations):

Config	Weight Bits	Activation Bits
W4A4	4	4
W4A8	4	8
W5A5	5	5
W6A6	6	6
W6A8	6	8
W8A8	8	8

Weight quantization uses per-channel granularity. Activation quantization uses per-token granularity for language models.

B.5. Distributed Training Infrastructure

DeepSpeed Configuration. We use DeepSpeed ZeRO Stage 2 for memory-efficient distributed training with the following settings:

- Gradient partitioning with allgather bucket size 2×10^8
- Reduce scatter with bucket size 2×10^8
- Communication overlap enabled
- Contiguous gradient buffers

Hardware. Experiments are conducted on 8 NVIDIA A100 GPUs with 80 GB memory each. We train both Llama and Qwen models for up to 10 epochs until convergence.

B.6. Evaluation Protocol

Language Model Benchmarks. We evaluate in a zero shot setting on HellaSwag, Winogrande, PIQA, TruthfulQA, WikiText, BoolQ, Arc-Easy and Arc-Challenge.

Vision Model Benchmarks. SigLIP models are evaluated on ImageNet-1K (Deng et al., 2009) zero-shot classification using the standard CLIP evaluation protocol with Top-1 and Top-5 accuracy metrics.

C. SigLIP Post-Training Quantization Results

Table 2 provides detailed results for SigLIP base 16-384 under post-training quantization with ERQ and RepQViT methods.

Model Configuration. We use the SigLIP base 16-384 vision encoder (Zhai et al., 2023), which processes images at 384×384 resolution with 16×16 patches. The model is evaluated on ImageNet-1K (Deng et al., 2009) classification using the partitioned test set (split 95-5).

Quantization Methods. We evaluate two PTQ methods designed for vision transformers:

- **ERQ** (Error-aware Quantization): A PTQ method that minimizes reconstruction error by considering the error propagation through transformer layers. ERQ optimizes quantization parameters to reduce the cumulative error in attention and feed-forward computations.
- **RepQViT** (Li et al., 2023): A reparameterization-based approach that decouples quantization scales for hardware-friendly deployment. RepQViT addresses the unique challenges of quantizing vision transformers by handling post-LayerNorm activations and attention score distributions.

Table 2. SigLIP base 16-384 PTQ results on ImageNet-1K zero-shot classification. All values are Top-1 accuracy (%). B: Baseline, B+N: Baseline with Jacobian-guided Noise. The FP32 baseline achieves 86.7% Top-1 accuracy.

Method	Noise	W4A4	W5A5	W6A6	W8A8
ERQ	B	51.2	72.4	81.3	85.9
	B+N	**60.1**	**76.8**	**83.5**	**86.4**
RepQViT	B	32.5	65.1	78.9	85.6
	B+N	**69.8**	**74.2**	**82.1**	**86.2**

Table 3. Qwen2.5-3B full-precision (FP16) results with and without noise injection. B: Baseline, B+N: With Jacobian Noise ($\alpha = 0.5$, rowwise).

Model	MMLU	HellaSwag	PIQA	Winogrande	TruthfulQA	WikiText PPL
Qwen2.5-3B (B)	65.25	75.08	79.87	70.72	47.41	10.60
Qwen2.5-3B (B+N)	65.48	74.91	79.54	70.17	47.47	10.46

Bit-width Settings. We evaluate four quantization configurations denoted as WxAy, where x is the weight bit-width and y is the activation bit-width:

- **W4A4**: Aggressive 4-bit quantization for both weights and activations, suitable for edge deployment with severe memory constraints.
- **W5A5**: Moderate 5-bit quantization offering a balance between compression and accuracy.
- **W6A6**: Conservative 6-bit quantization with minimal accuracy degradation.
- **W8A8**: Near-lossless 8-bit quantization serving as a reference point.

Training Protocol. Models are fine-tuned with Jacobian-guided noise injection as described in Section 2, using the hyperparameters specified in Section 3.1. The noise scaling coefficient α is set to 0.5 based on the sensitivity analysis in Figure 3.

D. Full-Precision Results With and Without Noise

Table 3 presents evaluation results for Qwen2.5-3B at full precision (FP16) with and without Jacobian-guided noise injection during training. This demonstrates that noise injection does not degrade full-precision model performance while providing quantization robustness.

E. Detailed zero-shot PTQ results for Llama and Qwen

Tables 4, 5 and 6 shows the detailed results for Llama and Qwen model on all 7 benchmarks. We observe that noise injections improves the performance of the overall model, while maintaining comparable or better performance across all 7 benchmarks.

F. Activation Similarity Analysis on Hardware

To validate our method in real-world deployment scenarios, we analyze a Vision-Language Model (VLM) trained with quantization-aware training at 15-bit precision and deployed on edge hardware. We measure the cosine similarity between quantized and unquantized activations at each layer, providing a direct measure of how well the quantized model preserves the computational behavior of the full-precision model.

Experimental Setup. The VLM is deployed on edge accelerator hardware with 15-bit fixed-point arithmetic. Unlike standard GPU deployment where numerical behavior closely matches training (bfloat16/float32), edge accelerators often have distinct rounding behavior that can cause divergent generation even at relatively high bit-widths.

Table 4. SpinQuant W4A4KV4 / W4A8KV8 / W6A6KV6 / W6A8KV8 full results for Qwen2.5-3B and Llama-3.2-3B on both the Base and Jacobian-noise (Jac) variants. PPL is WikiText-2. Zero-shot metric is `acc_norm` for HellaSwag/PIQA/ARC-C and `acc` otherwise. "Avg7" is the average of HellaSwag, PIQA, WinoGrande, ARC-E, ARC-C, BoolQ, TQA-MC2.

Model	Variant	Precision	PPL	HellaSwag	PIQA	WinoGrande	ARC-E	ARC-C	BoolQ	TQA-MC2	Avg7
Qwen2.5-3B	Base	W4A4KV4	10.49	66.27	74.32	59.19	71.93	44.20	65.32	41.12	60.34
		W4A8KV8	8.55	72.91	78.02	68.98	77.48	48.21	79.94	45.91	67.35
		W6A6KV6	8.36	73.92	78.35	67.25	77.15	48.38	79.08	46.64	67.25
		W6A8KV8	8.32	74.25	79.33	68.59	76.18	48.98	79.57	46.74	67.66
	Jac-noise	W4A4KV4	10.33	66.65	74.43	61.88	73.70	45.05	70.18	46.42	62.62
		W4A8KV8	8.35	72.32	78.40	68.35	74.03	45.73	77.55	43.25	65.66
		W6A6KV6	8.18	73.26	78.62	68.27	77.69	48.81	79.08	46.32	67.44
		W6A8KV8	8.11	73.56	78.89	67.64	77.10	49.40	79.54	46.03	67.45
Llama-3.2-3B	Base	W4A4KV4	10.72	66.96	74.05	60.38	69.99	39.76	71.71	38.89	60.25
		W4A8KV8	8.31	72.44	77.37	68.82	73.65	43.77	74.16	41.12	64.48
		W6A6KV6	8.03	73.74	78.67	69.46	76.47	46.59	75.29	41.66	65.98
		W6A8KV8	7.97	73.86	78.13	69.93	74.07	47.78	76.24	41.64	65.95
	Jac-noise	W4A4KV4	11.18	67.38	73.50	64.09	67.05	40.27	70.40	38.35	60.15
		W4A8KV8	8.47	73.33	77.97	68.35	74.71	46.33	73.85	41.65	65.17
		W6A6KV6	8.17	74.67	77.91	70.88	77.31	48.38	76.24	42.31	66.81
		W6A8KV8	8.12	74.94	78.35	70.01	75.13	49.66	76.27	42.70	66.72

Table 5. AWQ W4A4 / W4A8 / W6A6 / W8A8 results for Qwen2.5-3B and Llama-3.2-3B on both the Base and Jacobian-noise (Jac) variants. PPL is WikiText-2. Zero-shot metric is `acc_norm` for HellaSwag/PIQA/ARC-C and `acc` otherwise. "Avg7" is the average of HellaSwag, PIQA, WinoGrande, ARC-E, ARC-C, BoolQ, TQA-MC2.

Model	Variant	Precision	PPL	HellaSwag	PIQA	WinoGrande	ARC-E	ARC-C	BoolQ	TQA-MC2	Avg7
Qwen2.5-3B	Base	W4A4	8121.24	30.23	54.35	48.54	32.32	23.12	48.07	47.45	40.58
		W4A8	10.80	74.23	77.97	68.11	78.32	49.57	80.18	46.21	67.80
		W6A6	15.29	70.26	74.86	64.72	75.80	49.40	75.54	45.25	65.12
		W8A8	10.80	74.23	77.80	68.59	77.99	50.00	80.70	45.57	67.84
	Jac-noise	W4A4	3281.04	30.57	53.26	52.25	33.54	24.06	49.42	47.80	41.56
		W4A8	11.01	73.45	78.24	68.51	78.11	49.40	80.03	44.61	67.48
		W6A6	13.63	70.95	75.81	64.61	76.29	48.81	74.84	46.35	65.38
		W8A8	10.80	74.44	79.40	68.25	79.20	50.49	81.12	45.73	68.38
Llama-3.2-3B	Base	W4A4	105.06	32.47	55.17	51.14	33.71	23.81	50.80	47.31	42.06
		W4A8	9.96	74.58	78.24	71.03	77.06	47.61	76.39	41.64	66.65
		W6A6	10.80	72.87	76.99	68.35	74.45	45.05	70.58	40.21	64.07
		W8A8	9.73	74.42	78.62	70.96	77.10	47.95	76.27	41.04	66.62
	Jac-noise	W4A4	104.34	33.33	55.41	51.04	33.79	24.29	51.55	45.99	42.20
		W4A8	9.86	75.17	78.78	70.48	78.24	49.66	77.13	42.00	67.35
		W6A6	10.55	73.46	77.04	67.88	75.08	47.01	72.97	40.52	64.85
		W8A8	9.68	75.41	78.94	70.72	78.54	49.74	76.91	42.61	67.55

Results. Figure 5 reveals a striking difference between baseline and noise-trained models:

- **Baseline (a):** The cosine similarity between quantized and unquantized activations fluctuates dramatically across layers, with multiple layers showing similarity scores approaching zero. This indicates that quantization errors accumulate and amplify through the network, causing the quantized model's internal representations to diverge completely from the full-precision model.
- **With Noise Injection (b):** The noise-trained model maintains consistently high cosine similarity (>0.95) across all layers, demonstrating that Jacobian-guided noise injection trains the model to be inherently robust to numerical perturbations introduced by quantization.

We also reproduced this in a PTQ setting with W4A4 Round-to-Nearest (RTN) quantisation with Siglip base 384 model. Figure 6 shows that errors accumulate over layers and eventually result in divergence of activation values. However, training with noise reduces this divergence and helps model consistently maintain high similarity with actual activation values.

Implications for Deployment. The layer-wise similarity analysis provides direct evidence that softmax instability, as characterized in Section 2, causes real deployment failures. When similarity drops to near-zero at intermediate layers, the

Table 6. GPTQ W4A4 / W4A8 / W6A6 / W8A8 results for Qwen2.5-3B and Llama-3.2-3B on both the Base and Jacobian-noise (Jac) variants. PPL is WikiText-2. Zero-shot metric is `acc_norm` for HellaSwag/PIQA/ARC-C and `acc` otherwise. "Avg7" is the average of HellaSwag, PIQA, WinoGrande, ARC-E, ARC-C, BoolQ, TQA-MC2.

Model	Variant	Precision	PPL	HellaSwag	PIQA	WinoGrande	ARC-E	ARC-C	BoolQ	TQA-MC2	Avg7
Qwen2.5-3B	Base	W4A4	5172.39	25.81	49.95	51.14	26.52	23.46	43.91	49.27	38.58
		W4A8	14.13	72.05	77.15	65.98	78.00	48.89	80.03	44.43	66.65
		W6A6	28.42	63.32	70.78	60.46	69.15	42.66	68.23	45.19	59.97
		W8A8	10.98	73.99	78.02	67.88	77.99	49.15	80.06	45.78	67.55
	Jac-noise	W4A4	4207.85	26.31	50.05	51.04	26.00	24.32	45.38	49.64	38.96
		W4A8	12.21	71.39	77.42	66.77	77.23	49.40	77.43	43.14	66.11
		W6A6	21.52	63.10	71.98	60.77	69.28	43.86	69.20	42.54	60.10
		W8A8	10.82	73.41	78.65	68.23	78.95	48.91	80.36	44.73	67.61
Llama-3.2-3B	Base	W4A4	454.35	29.75	51.36	49.88	30.89	23.55	46.79	49.96	40.31
		W4A8	37.40	72.75	77.91	67.80	74.20	44.97	76.91	39.10	64.81
		W6A6	10.87	72.05	76.55	66.61	73.32	45.31	71.53	40.49	63.69
		W8A8	9.69	74.37	78.62	71.27	76.77	48.38	75.93	41.50	66.69
	Jac-noise	W4A4	496.38	30.84	54.21	48.67	30.76	25.91	46.87	47.77	41.00
		W4A8	31.47	73.53	78.13	70.96	76.85	47.61	76.64	40.82	66.36
		W6A6	11.12	73.19	76.66	66.46	74.96	45.48	71.56	40.81	64.16
		W8A8	9.86	75.21	78.56	69.53	78.07	49.40	77.37	42.64	67.25

error propagates and compounds through subsequent attention operations, leading to outputs that bear little resemblance to the full-precision model. Our noise injection method addresses this root cause by regularizing the attention mechanism during training, resulting in models that maintain consistent behavior when deployed on diverse hardware with varying numerical precision.

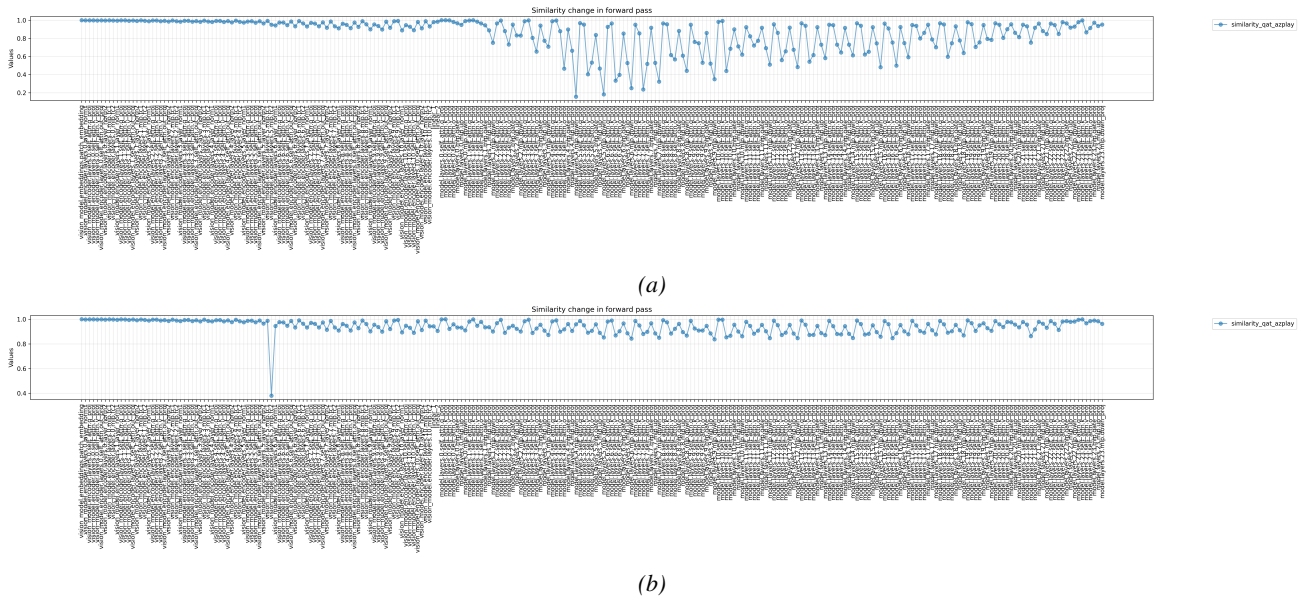


Figure 5. Layer-wise cosine similarity between quantized (15-bit) and unquantized activations for a VLM deployed on edge hardware. (a) Baseline model shows severe similarity degradation, with some layers dropping to near-zero cosine similarity. (b) Model trained with Jacobian-guided noise injection maintains consistently high similarity (>0.95) across all layers.

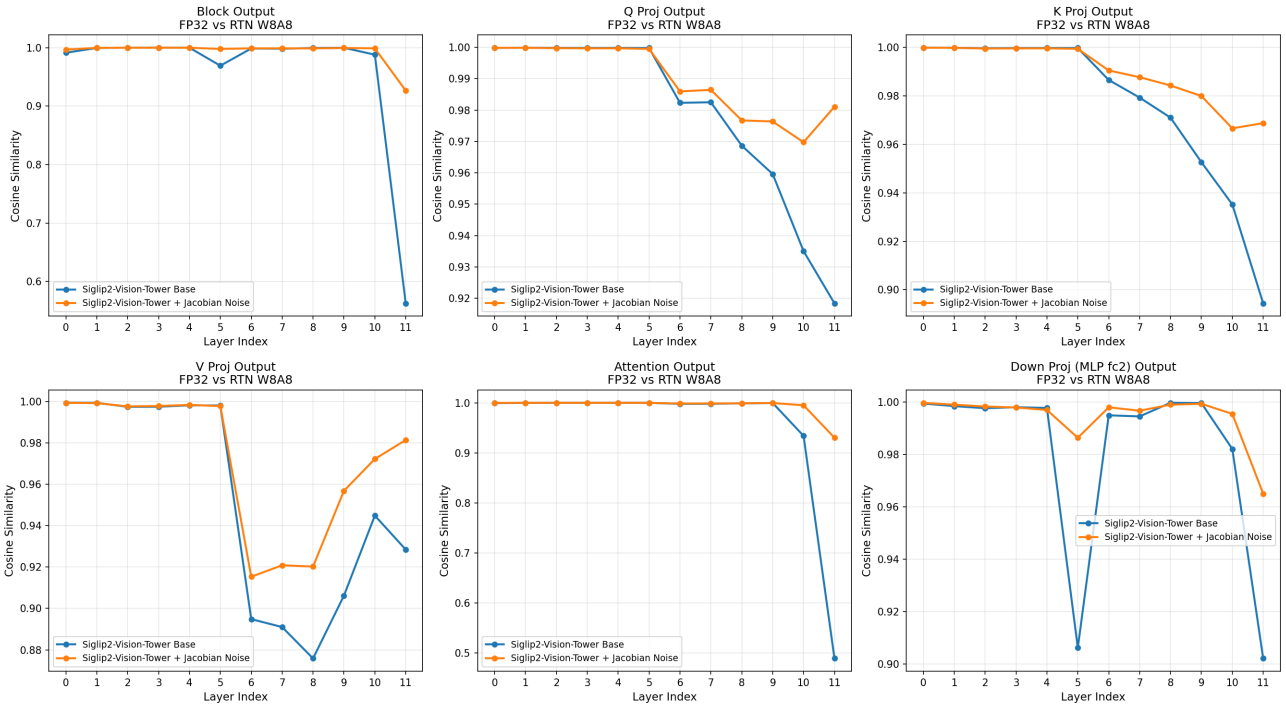


Figure 6. Attention maps for SigLIP (layer 10) Baseline vs baseline with noise. Noise-trained models exhibit more diffused attention patterns across key positions.