Exploring Layer-wise Information Effectiveness for Post-Training Quantization in Small Language Models

He Xiao[1], Qingyao Yang[1], Dirui Xie[2], Wendong Xu[1], Zunhai Su[1,3], Runming Yang[1], Haobo Liu[1], Wenyong Zhou[1], Zhengwu Liu[1], Ngai Wong[1]*

[1] The University of Hong Kong, [2] Huazhong University of Science and Technology
[3] Shenzhen International Graduate School, Tsinghua University

* Corresponding author: nwong@eee.hku.hk

Abstract

Large language models with billions of parameters are often over-provisioned: many layers contribute little unique information yet dominate the memory and energy footprint during inference. We present LieQ (Layer-wise information effectiveness Quantization), a hardware-native, metric-driven post-training quantization framework that addresses the critical challenge of maintaining accuracy in *sub-8B* models, model parameters less than 8B, under extreme low-bit compression. LieQ keeps *uniform bit-width within each layer* while mixing precision across layers, preserving standard multiplication kernels and avoiding irregular memory access, codebooks, or irregular formats at inference time. Our method uncovers a strong correlation between layer-wise *functional saliency* and *representational compactness*, revealing that layers with higher training-induced energy concentration are functionally irreplaceable. Leveraging this insight, we propose a purely *geometry-driven* sensitivity proxy that enables automatic bit-width allocation under a target average-bit budget without expensive gradient updates or inference-based perplexity probing. Under an *average* weight bit-width approaching two bits per parameter, LieQ consistently reduces the large accuracy gap typically observed for naive uniform 2-bit baselines on Qwen3 and LLaMA3.x families, while retaining standard-kernel efficiency. These properties make LieQ a practical path toward deploying small language models on resource-constrained edge devices. Code will be available at: https://github.com/HeXiao-55/LieQ-official.git.

1 Introduction

Large language models (LLMs) have achieved remarkable success across a variety of natural language processing tasks (Achiam et al., 2023; Guo et al., 2025), yet their immense parameter counts and activation sizes impose severe burdens on memory footprint and inference latency: a 7 billion (B) model fits into a single GPU, but 70 B variants already demand 140 GB of memory, far exceeding the less than 8 GB budgets of widely deployed edge devices. Robotics applications also require lower-power small language models (SLMs) to run on robots/cars/drones with a constrained memory budget. SLMs are the future of generic edge AI (Belcak et al., 2025). However, even models below 7 B parameters, often marketed as "mobile-friendly", still surpass the 4 to 12 GB memory envelopes of consumer phones and single-board computers, making aggressive compression an indispensable prerequisite for on-device inference (Grattafiori et al., 2024; Touvron et al., 2023; Yang et al., 2024; QwenTeam, 2025). This memory wall is not merely an engineering inconvenience; it is a fundamental barrier to democratizing LLMs.

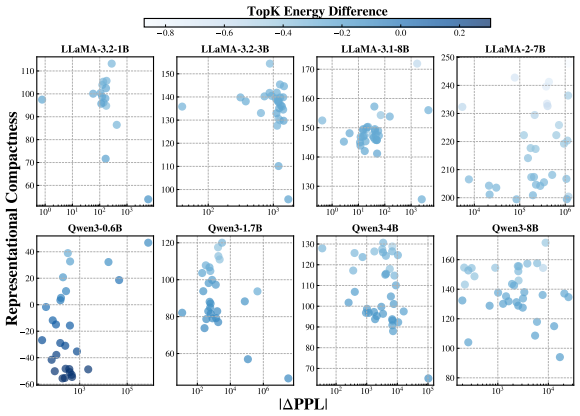

Figure 1: Layer-wise information taxonomy (each dot corresponds to one layer) across three correlated diagnostic metrics. Smaller models (e.g., Qwen-0.6B) exhibit lower robustness under extreme low-bit quantization, with certain layers being significantly more critical (clustered dots with deeper color) than others. Increasing the model size spreads out and balances the importance across layers.

Quantization techniques offer a pathway to compress model weights and activations into low-bit representations, but in practice they strug-

Findings of the Association for Computational Linguistics: ACL 2026, pages 15753–15764
July 2-7, 2026

gle to maintain the original model accuracy. Quantization-aware training (QAT) can alleviate this but with the cost of substantial retraining overhead (Chen et al., 2025). Post-training quantization (PTQ) has therefore emerged as a promising remedy, compressing weights to low-bit without retraining (Frantar et al., 2022; Lin et al., 2024). However, PTQ often causes severe performance degradation at ultra-low bit-widths, with 2-bit quantization increasing perplexity by two orders of magnitude. *This problem is especially severe in compact models (sub-8B parameters) due to their limited redundancy to absorb quantization noise.* Empirical studies confirm that the perplexity gap between full-precision and 2-bit representations narrows as the model size increases. In addition, existing low-bit approaches have limitations: sparse-outlier methods rely on heuristics for bit allocation (Dettmers et al., 2024); salience-driven quantization maintains uniform bit budgets across layers (Shang et al., 2023; Huang et al., 2024); and finer-grained methods like APTQ (Guan et al., 2024; Huang et al., 2025) and LLM-MQ (Li et al., 2023) sacrifice hardware efficiency with heterogeneous data formats. This raises our first research question: **Challenge 1.) Can we employ a structured PTQ scheme that preserves accuracy while maintaining a regular weight layout?**

Beyond precision loss, accuracy collapse also stems from the intrinsic model architecture. State-of-the-art (SOTA) LLMs predominantly follow the Transformer framework whose attention mechanism, while powerful, is notably fragile. Transformer layers exhibit heterogeneous saliency: some are highly vulnerable to precision loss, whereas others tolerate aggressive bit reduction. Prior work relies on costly "drop-one-layer" perplexity probing (functional saliency) to identify these critical layers. However, this functional behavior must stem from the underlying weight structures. This raises a fundamental question: **Challenge 2.) Can we identify a static geometric signature and theoretical guidance that explains and predicts functional saliency without pre-probing?**

Finally, extreme low-bit quantization (2-bit) must confront memory-budget heterogeneity and hardware-efficiency tension. Edge accelerators, mobile SoCs and existing GPUs impose diverse memory ceilings, so a practical scheme must guaranty a target footprint while maximizing retained capability. Although per-vector or per-element mixed precision achieves high fidelity, it introduces irregular layouts, extra indices and costly decoding steps that offset theoretical gains (Huang et al., 2025, 2024; Shang et al., 2023). Codebook-based 2-bit methods also add runtime transforms that complicate deployment (Egiazarian et al., 2024; Ashkboos et al., 2024; Chee et al., 2023; Shao et al., 2024). Hence, we ask: **Challenge 3.) How can we simultaneously attain hardware efficiency and accuracy robustness under extreme low-bit PTQ, while preserving standard kernels?**

Therefore, targeting these challenges, we present LieQ (Layer-wise information effectiveness Quantization), a principled PTQ framework. LieQ rests on a key analytical finding: **a layer's functional indispensability is strongly correlated with its representational compactness change.** We show that layers which undergo significant entropy reduction, learning to concentrate information into lower-rank manifolds, are the ones that cannot tolerate quantization noise. This insight allows us to use static geometric properties as a reliable proxy for saliency. Therefore, LieQ presents a main conclusion: **use high-precision(4-bit) for the layer with the most significant geometric feature (most compactness) and lower(2-bit) for others can retain performance to the maximum extent.** Our method dynamically allocates precision according to this geometric proxy, concentrating high-precision bits where they matter most while preserving hardware-friendly *uniform* within-layer layouts and standard multiplication kernels. Our key contributions are:

1. We conduct a rigorous study connecting layer-wise functional saliency with representational geometry. We demonstrate that *Compactness Shift*, the reduction in singular value entropy, serves as a strong predictor of a layer's true importance, offering an interpretability perspective on why certain layers break under quantization.

2. We propose a purely geometry-driven quantization schedule. Instead of running expensive perplexity evaluations or tuning hyperparameter weights, LieQ selects high-precision layers solely based on their distribution properties. This yields a *probing-free layer selection process* that is mathematically elegant and robust.

3. We link this geometric ranking to a closed-form budget rule, achieving SOTA accuracy at an *average* bit-width of $\sim$2.0 bits per weight on Qwen3 and LLaMA3.x families while maintaining uniform-within-layer regularity for standard kernel compatibility.

2 Related Works

Mixed-precision PTQ for Compact LLMs. Early efforts such as GPTQ and RPTQ (Frantar et al., 2022; Yuan et al., 2023) reduced 175 B models to 4 bit with modest degradation, yet later studies revealed that the same routines collapse on sub-7B models where redundancy is scarce (Li et al., 2025). Subsequent methods therefore inject protective mechanisms: outlier sparsity (Dettmers et al., 2024), salience-driven grouping with mixed precision (Huang et al., 2025), and second-order error metrics with finer-grained mixed-precision (Guan et al., 2024; Li et al., 2023). While these techniques restore language modeling accuracy at 2 to 4-bit, they typically rely on non-uniform formats that break tensor contiguity and hinder kernel fusion, motivating our search for a structured yet hardware-friendly scheme.

Weights Effectiveness Diagnostics. Transformer layers exhibit heterogeneous saliency: some are highly vulnerable to precision loss, whereas others tolerate aggressive bit reduction. To decide which layers deserve higher precision, prior work estimates sensitivity via weight reconstruction error (Dong et al., 2019), Hessian eigenvalues (Yao et al., 2021), activation entropy (Behtash et al., 2025), or direct layer-wise quantization (Dumitru et al., 2024). Previous studies on systematic outliers (An et al., 2025; Su and Yuan, 2025) in LLMs have shown that certain shallow layers generate function-specific outliers and are particularly sensitive to model compression techniques (Yu et al., 2024; Su et al., 2025). Recent probes measure representational geometry, i.e., rank expansion and spectral concentration (Wei et al., 2024), offering an alternative view to loss and accuracy.

Bit-Width Allocation Algorithms. Given per-layer scores, the remaining task is to satisfy a global memory budget based on hardware constraints. Greedy heuristics dominate early mixer-precision quantization papers, but their myopic choices often miss the ideals. Integer inference (Jacob et al., 2017) and differentiable search (Ma et al., 2025) achieve better solutions at higher cost, whereas hardware-aware schedulers such as HAWQ-v3 (Yao et al., 2021) trades optimality for throughput. Our approach inherits the efficiency of greedy search yet benefits from more informative scores, yielding a closed-form allocation for typical 2/3/4-bit settings.

Extremely Low-Bit Quantization. QuIP and QuIP# (Chee et al., 2023; Tseng et al., 2024) introduce rotations (Hadamard) to improve incoherence and enable 2-bit quantization, while QTIP (Tseng et al., 2025) leverages trellis coding and AQLM (Egiazarian et al., 2024) uses additive codebooks for extreme compression. These methods can deliver strong 2–3 bit accuracy but typically require additional process or runtime transforms and irregular memory access, complicating deployment. PTQTP (Xiao et al., 2025) develops a more efficient ternary implementation of robust and plug-in PTQ method while performing impressive performance on mainstream datasets. They offer higher fidelity at moderate bit-widths, often trading hardware regularity and additional operations for accuracy. QuaRot (Ashkboos et al., 2024) learns rotations to remove outliers for 4-bit, and SpQR (Dettmers et al., 2024) combines sparsity and quantization to approach nearly lossless compression. In contrast, our approach targets the complementary regime of near-2-bit budgets while preserving standard multiplication kernels via uniform-within-layer layouts and mixes precision only across layers for ease of deployment.

3 Explore Stably Efficient PTQ Criterion

The research landscape of efficient and robust PTQ can be mapped on solving the challenges highlighted in *C1-C3* from three axes: (i) accuracy-compression trade-offs in *sub-8B* PTQ, (ii) layer-wise diagnostics for guiding mixed precision, and (iii) automatically bit-width allocation under hardware constraints.

3.1 Sensitivity Diagnostics

Existing methods either secure accuracy at the expense of irregular layouts or retain regularity while ignoring layer heterogeneity. To address these challenges, we introduce a principled metric-driven quantization strategy that quantifies the saliency of each layer via complementary diagnostics.

Functional Saliency. Our goal is to quantify the unique information contributed by each Transformer layer in an auto-regressive model $\mathcal{M}$ composed of L total layers. Given a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ containing N token sequences of length-

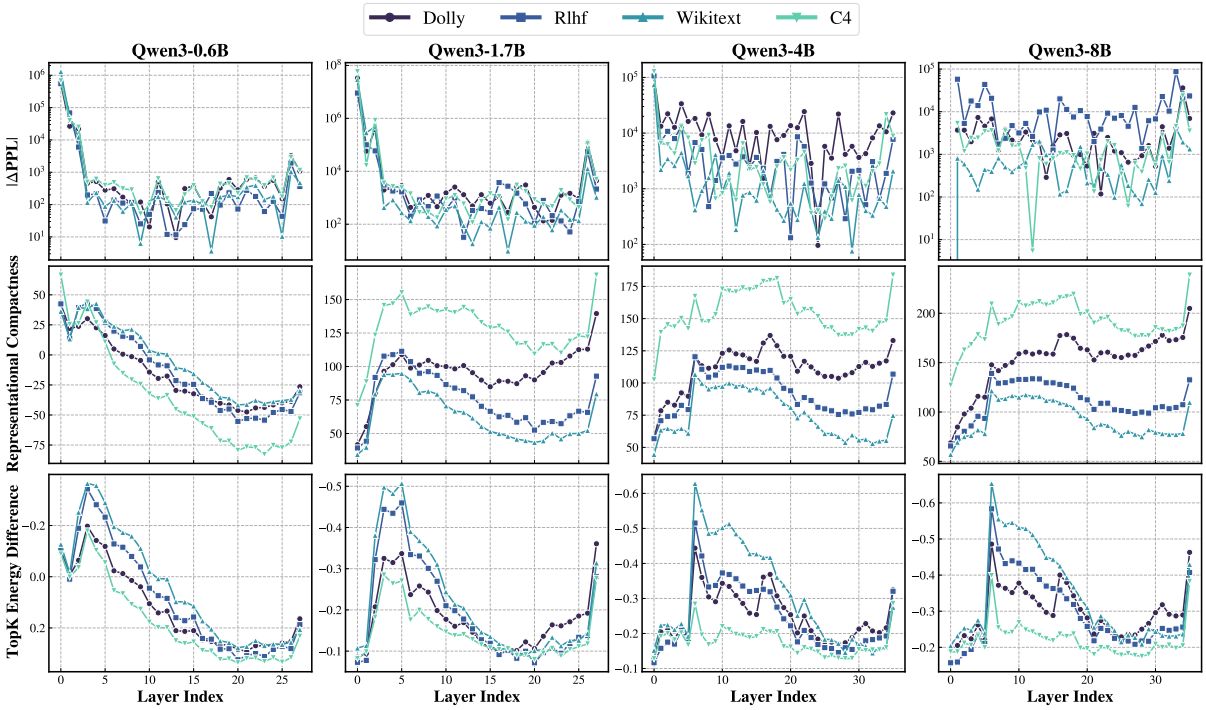

Figure 2: Functional diagnostic measures the drop in perplexity when a layer is removed on Qwen3 family. We use the representational compactness and TopK energy to proxy the significant perplexity loss.

T, we define the baseline negative log-likelihood

$$\mathcal{L}_{\text{base}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\log p_{\mathcal{M}}(x_t^{(i)} \mid x_{<t}^{(i)}) \quad (1)$$

Where $p_{\mathcal{M}}(x_t^{(i)} \mid x_{<t}^{(i)})$ is the probability of a token x_t given the preceding tokens $x_{<t}$. The exponent of this loss yields the baseline perplexity, $\text{PPL}_{\text{base}} = \exp(\mathcal{L}_{\text{base}})$. By replacing the ℓ-th Transformer block with an identity mapping plus its residual projection, we obtain the perturbed model $\mathcal{M}_{\setminus \ell}$ and record its perplexity shift:

$$\Delta\text{PPL}_\ell = \text{PPL}_{\setminus \ell} - \text{PPL}_{\text{base}} \quad (2)$$

Where $\text{PPL}_{\setminus \ell}$ is the perplexity of the model without layer ℓ. This metric directly measures the drop in predictive performance attributable to the absence of layer ℓ. However, computing ΔPPL requires $(L+1)N$ forward passes, making it computationally prohibitive for on-device or rapid deployment scenarios. This motivates the search for a geometric proxy.

Finding 1. Smaller models (e.g., Qwen-0.6B) exhibit lower robustness under extremely low-bit quantization (∼2-bit), with certain important layers standing out as significantly more critical than others. As illustrated in Figure 1, increasing the model size leads to a more uniform distribution of representational compactness, resulting in a more balanced importance across layers. However, this also makes it harder to distinguish the relative significance of each layer.

Representational Compactness. We introduce a geometric proxy that quantifies layer saliency via Representational compactness properties. The theoretical foundation rests on the hypothesis that training enhances layer effectiveness by concentrating meaningful information into structured manifolds, which can be detected through changes in the singular value distribution. For each type of linear projection P in the Transformer architecture, we analyze the projected representations $Z = W_P^{(\ell)}\mathbf{h}^{(\ell)} \in \mathbb{R}^{T\times d_{\text{head}}}$. Here, $W_P^{(\ell)} \in \mathbb{R}^{d_{\text{head}}\times d}$ represents the learned weight matrix for the respective projection type P in layer ℓ, transforming the input hidden states $\mathbf{h}^{(\ell)} \in \mathbb{R}^{T\times d}$ into the corresponding space of attention head. To establish a baseline for comparison, we generate a randomly-initialized counterpart $\tilde{Z} = \tilde{W}_P^{(\ell)}\mathbf{h}^{(\ell)}$, where $\tilde{W}_P^{(\ell)}$ follows the same initialization distribution as the original weights but remains untrained.

The core insight is that training should increase the effectiveness of the layer by organizing information into tasks-relevant structures while eliminating

noise and redundancy. To quantify this, we perform singular value decomposition on both matrices:

$$Z = U\Sigma V^T; \quad \tilde{Z} = \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{3}$$

yielding singular values $\{\sigma_k\}$ and $\{\tilde{\sigma}_k\}$, respectively. We then compute the representational compactness, defined as the exponential of the Shannon entropy of the energy distribution:

$$\text{Compact}(Z) = \exp\Big(-\sum_{k=1}^{K} p_k \log p_k\Big) \tag{4}$$

$$p_k = \frac{\sigma_k^2}{\sum_{j=1}^{K} \sigma_j^2} \tag{5}$$

Here, p_k represents the normalized energy of the k-th singular value in $K = \min(T, d_{\text{head}})$ singular values. Representational compactness provides a smooth, differentiable measure of information concentration: when singular values are uniformly distributed, $\text{Compact}(Z)$ is high (indicating redundant representations), but when a few singular values dominate, $\text{Compact}(Z)$ is low (suggesting concentrated, sensitive representations). The layer effectiveness metric is then defined as the relative change in representational compactness:

$$\Delta r_\ell^{(P)} = \frac{\text{Compact}(\tilde{Z}) - \text{Compact}(Z)}{\text{Compact}(\tilde{Z})} \tag{6}$$

The normalization by $\text{Compact}(\tilde{Z})$ casts the metric as a relative change, making it a stable and comparable measure across different layers. It quantifies the proportional reduction in representational randomness, thus isolating the structural changes induced by training from baseline properties. A positive $\Delta r_\ell^{(P)}$ indicates that training has increased layer effectiveness by developing more concentrated representations, making the layer more critical for model performance.

Supporting Metric: Top-k Energy. While compactness captures the overall distributional concentration, we also inspect the top-k energy fraction to confirm information concentration:

$$E_k(Z) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_j \sigma_j^2}, \quad \Delta E_{k,\ell}^{(P)} = E_k(Z) - E_k(\tilde{Z}) \tag{7}$$

A positive $\Delta E_{k,\ell}^{(P)}$ signifies that training has shifted more energy into the dominant components. From our analysis, ΔE closely tracks Δr and serves as a secondary validation of the geometric shift. In our evaluation, we assess correlations on four diverse datasets: WikiText2 (Merity et al., 2016), Dolly-15k (Conover et al., 2023), HH-RLHF (Bai et al., 2022), and C4 (Raffel et al., 2020). We observe strong Spearman correlation ($\rho > 0.8$) between ΔPPL_ℓ and Δr_ℓ across depths, confirming that geometric compactness is a robust proxy for functional sensitivity.

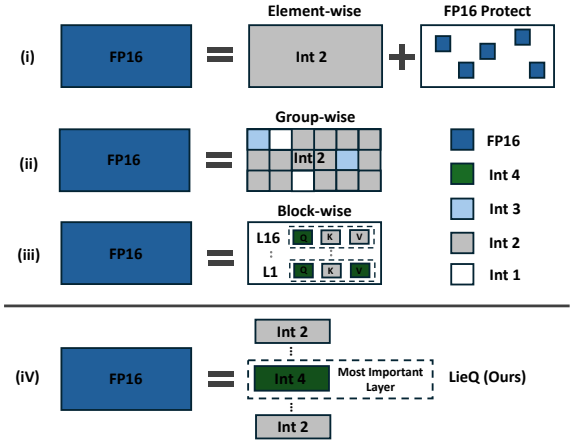

Figure 3: Illustration of the mixed-precision schemes. (i) Element-wise quantization with FP16 weights protection. (ii) Group-wise 2-bit quantization with 1-bit and 3-bit weights to balance accuracy and memory footprint. (iii) Block-wise 4-bit quantization within attention blocks in different layers. (iv) LieQ: Only one most significant layer with the most compact information is quantized to 4-bit, while the rest are quantized to 2-bit.

Finding 2. While different model families display distinct distributions of effective information, models within the same family consistently exhibit highly similar patterns across various test datasets, as shown in Figure 2. This intra-family consistency is observed across all three proposed metrics.

Additionally, we observe that shallow layers tend to be more important, consistent with prior studies on super weights (Yu et al., 2024; Su et al., 2025). Prior research has shown that certain shallow layers contain super weights, giving rise to systematic outliers (An et al., 2025) that reflect their mechanistic significance. In both the Qwen and LLaMA series, our metrics consistently reveal the key functional importance of super weight layers, aligning with known mechanistic effects.

3.2 Structured and Hardware-Friendly Mixed-Precision Quantization

Given the strong correlation between functional saliency and representational geometry, we dispense with perplexity probing and complex

Qwen3		Wiki				C4			
Weight Precision	**Method**	**0.6B**	**1.7B**	**4B**	**8B**	**0.6B**	**1.7B**	**4B**	**8B**
FP 16	-	20.9	16.7	13.64	9.71	30.31	22.36	19.83	15.41
2bit	GPTQ	$\underline{2.38 \times 10^4}$	$\underline{6.55 \times 10^2}$	5.92×10^5	7.77×10^4	1.70×10^6	3.08×10^6	4.37×10^5	5.46×10^5
	AWQ	1.21×10^7	7.52×10^6	1.38×10^7	1.21×10^7	1.03×10^9	7.70×10^8	3.18×10^8	1.37×10^{10}
	OmniQuant	2.55×10^5	N/A	5.30×10^4	7.31×10^4	$\underline{1.78 \times 10^5}$	N/A	5.81×10^5	9.27×10^4
	PB-LLM	1.52×10^5	9.27×10^5	4.68×10^3	$\underline{1.80 \times 10^3}$	2.03×10^5	$\underline{2.21 \times 10^6}$	$\underline{7.76 \times 10^3}$	$\underline{2.28 \times 10^3}$
	SliM-LLM	2.91×10^4	4.41×10^4	$\underline{39.71}$	N/A	N/A	N/A	N/A	N/A
	LieQ	**36.19**	**23.48**	**18.48**	**11.46**	**43.44**	**28.45**	**26.72**	**18.05**
3bit	AWQ	$\underline{2.20 \times 10^2}$	$\underline{84}$	$\underline{91}$	$\underline{27.5}$	$\underline{45.93}$	$\underline{27.15}$	$\underline{32.93}$	$\underline{17.38}$
	OmniQuant	1.54×10^5	N/A	1.95×10^5	4.59×10^4	2.09×10^5	N/A	3.42×10^5	4.73×10^4
	PB-LLM	2.47×10^4	1.35×10^5	3.23×10^3	1.23×10^3	3.03×10^4	1.43×10^5	2.28×10^3	1.09×10^2
	LieQ	**26.02**	**20.25**	**18.19**	**10.72**	**36.89**	**25.15**	**25.61**	**17.00**

Table 1: Zero-shot perplexity (lower is better) on WikiText-2 and C4 across Qwen3 model sizes. Large values use powers of ten ($m \times 10^k$, mantissa to two decimals). **Bold**: best; underlined: second-best among methods. "N/A": not reported or not applicable.

LLaMA3		Wiki			C4		
Weight Precision	**Method**	**1B**	**3B**	**8B**	**1B**	**3B**	**8B**
FP 16	-	9.75	7.80	6.23	14.01	11.33	9.53
2bit	GPTQ	4.97×10^3	2.42×10^3	7.17×10^2	1.96×10^4	$\underline{2.69 \times 10^2}$	$\underline{12.61}$
	AWQ	1.64×10^5	1.11×10^5	7.98×10^5	1.78×10^7	8.47×10^5	1.89×10^6
	OmniQuant	7.71×10^2	$\underline{5.38 \times 10^2}$	2.08×10^3	2.10×10^3	1.36×10^3	4.10×10^3
	PB-LLM	3.87×10^3	1.01×10^4	1.89×10^3	1.86×10^3	1.98×10^3	1.17×10^3
	SliM-LLM	$\underline{3.31 \times 10^2}$	8.19×10^2	$\underline{31.52}$	$\underline{6.92 \times 10^2}$	2.90×10^3	1.79×10^2
	LieQ	**14.53**	**9.78**	**8.15**	**20.70**	**14.75**	**13.18**
3bit	GPTQ	18.99	16.34	9.12	**19.01**	$\underline{14.08}$	**10.18**
	AWQ	34.81	13.22	10.80	23.77	15.23	12.57
	OmniQuant	$\underline{15.79}$	$\underline{9.91}$	$\underline{8.03}$	25.49	15.98	13.59
	PB-LLM	1.42×10^3	4.97×10^2	1.07×10^3	8.84×10^2	3.48×10^2	6.45×10^2
	LieQ	**13.58**	**9.21**	**7.24**	$\underline{19.69}$	**13.86**	$\underline{11.41}$

Table 2: Zero-shot perplexity (lower is better) on Wiki (WikiText-2) and C4 across LLaMA3 models. Large values use powers of ten ($m \times 10^k$, mantissa to two decimals). **Bold**: best result; underlined: second-best.

weighted metrics. Instead, we propose **LieQ**, a purely geometry-driven quantization framework.

Geometry-Driven Layer Selection. We aggregate the compactness diagnostic over linear projections to obtain a single score per layer:

$$s_\ell = \Delta r_\ell = \mathbb{E}_P \, \mathbb{E}_{\text{head}} \big[\Delta r_\ell^{(P,\text{head})} \big] \tag{8}$$

This score s_ℓ represents the "geometric irreplaceability" of layer ℓ. We rank layers by descending s_ℓ to select the set of high-precision layers $\mathcal{S}_{\text{hi}}$ and low-precision layers $\mathcal{S}_{\text{lo}}$:

$$\mathcal{S}_{\text{hi}} = \underset{m}{\text{TopK}}(s_1, \ldots, s_L), \quad \mathcal{S}_{\text{lo}} = [L] \setminus \mathcal{S}_{\text{hi}} \tag{9}$$

When $\ell \in \mathcal{S}_{\text{hi}}$, we assign $b_\ell = 4$; when $\ell \in \mathcal{S}_{\text{lo}}$, $b_\ell = 2$. This selection is **data-free** in the sense that it requires no validation set inference, relying only on the intrinsic weight statistics, i.e., via SVD on a single representative forward pass.

Automatically Slipping Precision. Given a target average-bit budget $\bar{b} \in [2, 4]$, the fraction of 4-bit layers under equal-sized layers is $f = \frac{\bar{b}-2}{4-2}$, yielding a closed-form $m = \text{round}(fL)$. With unequal parameter counts N_ℓ, we select the smallest $\mathcal{S}_{\text{hi}}$ (by descending s_ℓ) such that $\sum_{\ell \in \mathcal{S}_{\text{hi}}} N_\ell \geq f \sum_{\ell=1}^{L} N_\ell$. Finally, we report the weights-only relative memory fraction w.r.t. FP16. Let N_ℓ denote the number of parameters in layer ℓ, thus the compression ratio (CR) can be defined as below:

$$\text{CR} = \frac{\sum_{\ell=1}^{L} b_\ell N_\ell}{16 \sum_{\ell=1}^{L} N_\ell}, \tag{10}$$

so that memory reduction (weights-only) relative to FP16 is $16/\bar{b}$ where $\bar{b} = \frac{\sum_\ell b_\ell N_\ell}{\sum_\ell N_\ell}$.

Integration with Existing Quantization Methods. LieQ is orthogonal to the choice of PTQ back-end, providing a plug-and-play guideline for exist PTQ methods. In our implementation, mixing the proposed diagnostic strategy with GPTQ-4bit or AWQ-4bit yields further compression while maintaining high accuracy. Although LieQ supports any budget-implied number of promoted layers, in order to probe the quantization limits we promoted the smallest subset (e.g., top-1 s_ℓ to 4-bit), quantizing the rest to 2-bit. This extreme configuration illuminates the minimum precision required to preserve model performance while maximizing compression, and it preserves standard multiplication kernels at inference time.

4 Experiments and Results

Evaluation Protocol. We implemented our evaluation on PyTorch using models from HuggingFace (Paszke et al., 2019). No task-specific calibration, or post-training fine-tuning was applied in any of our experiments. All of our experiments were conducted on a single NVIDIA RTX 3090 (24 GB) GPU with mixed precision enabled and gradient checkpointing disabled to preserve activation needed for rank analysis. We follow AWQ (Lin et al., 2024) acceleration setting to establish the evaluation of end-to-end performance, the sequence length are settled to 512 for peer comparison.

Models and Baselines. We evaluated the layer-wise information effectiveness of multiple mainstream LLM families including Qwen3 (Q3) (QwenTeam, 2025), LLaMA3.x (L3) (Grattafiori et al., 2024), and LLaMA1&2 (L1&L2) (Touvron et al., 2023). All evaluated model sizes are less than 7–8 B parameters. We compare against representative PTQ methods: SliM-LLM (Huang et al., 2025), AWQ (Lin et al., 2024), GPTQ (Frantar et al., 2022), and OmniQuant (Shao et al., 2024). We additionally position against rotation/codebook-based approaches reported in the literature, including QuIP/QuIP# (Chee et al., 2023; Tseng et al., 2024) and AQLM (Egiazarian et al., 2024), noting their differing deployment characteristics.

Evaluation Datasets and Metrics. For language modeling quality, we report perplexity on WikiText-2 (Merity et al., 2016) and C4 (Raffel et al., 2020). To assess cross-domain generalization and language reasoning performance, we evaluate LieQ and existing methods on ARC-C/E (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), Winogrande (Sakaguchi et al., 2021), and MMLU (Hendrycks et al., 2021), following established protocols (Gao et al., 2024).

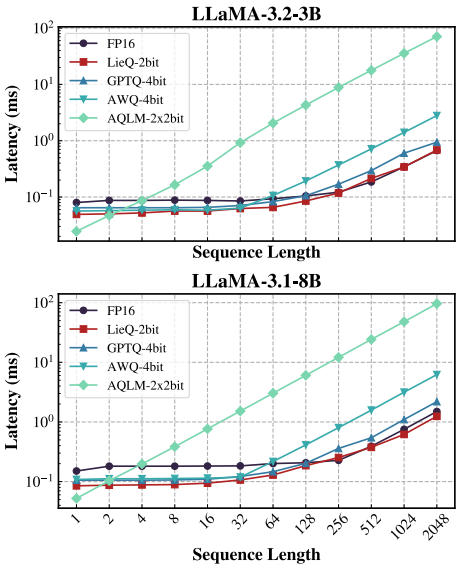

Figure 4: Microbenchmark latency of the gate_proj layer for LLaMA-3.2-3B and LLaMA-3.1-8B.

Results and Analysis. The primary aim of LieQ is to provide solid analysis and clear guidelines for structured extreme low-bit PTQ. In addition, using our proposed diagnostic methods, we have observed two key performance advantages of LieQ: **i) Competitive accuracy near two-bit *average* budgets.** As illustrated in Tables 1 and 2 and summarized in Table 3, LieQ substantially mitigates the degradation seen in naive 2-bit baselines on both perplexity and zero-shot tasks for Qwen3 and LLaMA 3.x, and achieves SOTA performance. While several 3-bit methods could reach higher peak accuracy in some settings, LieQ attained competitive results without introducing runtime rotations or codebooks. **ii) Hardware-friendly regularity.** With uniform bit-width *within* each layer, tensors remain contiguous and a single multiplication kernel per layer suffices. Figure 4 presents a per-layer microbenchmark on the gate projection of LLaMA-3.2-3B and LLaMA-3.1-8B, indicating latency reduction relative to FP16 under identical experimental conditions. Our focus here on acceleration is to preserve standard kernels and avoid irregular formats that fragment GPU through-

Model	Weight Precision	Method	PIQA	ARC-e	ARC-c	BoolQ	HellaSwag	Winogrande	MMLU
Q3-4B	FP16	-	74.97	80.47	50.51	85.14	52.24	65.82	68.25
	2.00	GPTQ	52.18	25.55	22.53	38.27	25.43	45.78	26.44
	2.00	AWQ	53.75	25.76	23.04	37.83	25.74	50.43	26.89
	2.00	OmniQuant	52.77	26.3	22.1	37.82	25.93	50.11	N/A
	2.00	SliM-LLM	57.51	39.39	19.62	56.45	31.29	49.25	N/A
	2.05	**LieQ**	**71.89**	**75.55**	**45.31**	**84.43**	**46.5**	**64.24**	**63.89**
Q3-4B	3.00	GPTQ	53.05	26.22	21.93	43.06	25.7	50.68	25.05
	3.00	AWQ	**72.03**	69.15	39.93	79.66	**47.11**	60.38	61.42
	3.00	OmniQuant	53.37	23.75	20.56	37.82	25.95	50.59	N/A
	3.00	**LieQ**	71.27	**71.84**	**42.15**	**84.53**	46.85	**64.88**	**65.92**
L2-7B	FP16	-	78.07	76.39	43.52	77.1	57.15	67.25	41.85
	2.00	GPTQ	58.71	38.01	22.44	50.98	29.93	53.43	23.26
	2.00	AWQ	49.73	26.53	20.99	37.83	26.14	49.8	24.51
	2.02	QuIP#	71.38	55.56	28.84	N/A	42.94	62.43	N/A
	2.02	AQLM	74.76	63.68	32.76	N/A	49.55	65.67	N/A
	2.29	AQLM	74.92	66.5	34.9	N/A	50.88	62.43	N/A
	2.05	**LieQ**	**77.48**	**68.1**	**38.14**	**69.45**	**53.75**	**65.98**	**32.57**
L2-7B	3.00	GPTQ	76.01	**72.9**	**40.44**	**74.5**	54.2	**67.96**	33.44
	3.00	AWQ	76.66	52.65	38.82	67.58	**70.66**	65.43	32.78
	3.04	AQLM	76.88	65.06	38.4	N/A	54.12	63.54	N/A
	3.00	**LieQ**	**77.31**	67.93	39.25	70.7	55.01	65.98	**34.39**
L3-3B	FP16	-	76.82	74.41	42.58	72.57	55.34	69.06	54.08
	2.00	GPTQ	53.37	26.94	20.65	44.56	26.81	50.28	23.66
	2.00	AWQ	52.61	24.79	22.27	37.83	25.38	49.64	26.89
	2.00	OmniQuant	55.27	29.54	18.6	37.83	26.3	50.27	N/A
	2.00	SliM-LLM	53.48	26.43	19.71	38.62	26	50.75	N/A
	2.07	**LieQ**	**75.41**	**70.33**	**37.71**	**64.5**	**50.88**	**68.11**	**48.25**
L3-3B	3.00	GPTQ	72.85	67.63	36.6	65.5	51.39	64.93	41.28
	3.00	AWQ	74.1	65.74	36.69	72.32	50.13	66.33	46.51
	3.00	OmniQuant	73.61	65.06	34.38	67.33	49.39	63.3	N/A
	3.00	**LieQ**	**75.35**	**72.18**	**40.19**	**72.32**	**52.34**	**66.93**	**49.62**

Table 3: Comparison (higher is better) on seven zero-shot reasoning tasks. **Bold**: best; underlined: second-best. "N/A": not reported or not available in the original source.

put. Moreover, we implemented and evaluated the method end-to-end on the Alpaca dataset using open-source implementations; the results are presented in Table 4.

Models	Baseline (Token/s)	Baseline (GB)	**LieQ** (Token/s)	**LieQ** (GB)
L1-7B	38.85	12.68	50.89	4.32
L1-13B	OOM	26.03	40.09	7.83
L2-7B	39.73	12.86	50.67	4.32

Table 4: End-to-end decode speed evaluation.

Discussion The gap between small and large models under LieQ is explained by how *redundancy* is distributed across layers, not by a limitation of the method. As in Figure 1, smaller models show concentrated criticality: a few layers exhibit large compactness shifts (deep clusters), so a "protect few, quantize the rest" schedule is essential. In larger models (e.g., 8B+), layer-wise compactness becomes more uniform, so fixing a small fraction of layers at higher precision yields diminishing returns. Heuristically, the cross-layer variance of the compactness shift Δr_ℓ scales inversely with model width: heterogeneity is higher in the sub-8B regime where LieQ is most advantageous.

Sensitivity to Bit Budget. Our experimental results demonstrate that LieQ maintains both excellent accuracy and inference performance on SLMs when using the structured, hardware-friendly mixed-precision framework. Furthermore, to explore the trade-off between the number of high-precision layers and model performance, we adjusted the budget configurations to conduct the ablation study on the automatic bit allocation method. Figure 5 shows the average language reasoning performance as we increased the number of 4-bit quantized layers (selected by our geometric proxy)

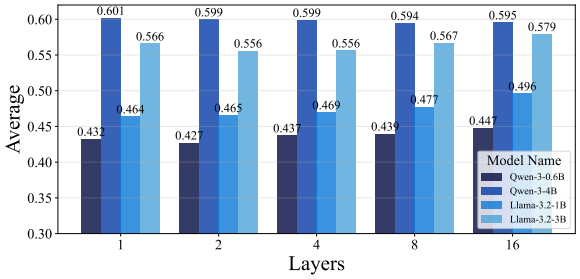

Figure 5: Average accuracy difference on language reasoning tasks with various precision configurations on small language models.

from 1 to 16; this range covers precisions from 2-bit to 3-bit levels. This confirms that protecting even the smallest subset (one layer) of geometrically critical layers yields a significant recovery in accuracy.

5 Conclusion

In this work, we first identified a strong correlation between a layer's functional saliency and its representational geometry, revealing that layers with high energy concentration are structurally irreplaceable. Building on this discovery, we proposed LieQ, a hardware-native PTQ framework that utilizes this geometric property as a proxy for bit-width allocation. Our experiments on Qwen3 and LLaMA3.x confirm that LieQ effectively identifies critical layers, substantially mitigating the accuracy collapse of naive 2-bit baselines while preserving hardware efficiency. These results not only offer a practical compression tool but also suggest that quantization sensitivity is a fundamental structural property rooted in weight manifolds, rather than a stochastic phenomenon.

Limitations

Our goal is to establish a analysis proxy for guiding extremely low-bit quantization. Therefore, although we have evaluated inference efficiency via a simple end-to-end estimate, significant room remains for optimization in practical engineering. End-to-end inference throughput optimization will depend on specific system-level implementations. Nevertheless, we believe this exploration path is applicable to LieQ (thanks to our simple, symmetrical core design) and we will pursue further optimization in future engineering work.

Acknowledgments

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council of Hong Kong, and in part by the AVNET-HKU Emerging Microelectronics and Ubiquitous Systems (EMUS) Lab.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2025. Systematic outliers in large language models. *arXiv preprint arXiv:2502.06415*.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Preprint*, arXiv:2404.00456.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Alireza Behtash, Marijan Fofonjka, Ethan Baird, Tyler Mauer, Hossein Moghimifam, David Stout, and Joel Dennison. 2025. Universality of layer-level entropy-weighted quantization beyond model architecture and size. *Preprint*, arXiv:2503.04704.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36:4396–4429.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2025. Efficientqat: Efficient quantization-aware training for large language models. *Preprint*, arXiv:2407.11062.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Tim Dettmers, Ruslan A. Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2024. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. In *The Twelfth International Conference on Learning Representations*.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. 2019. Hawq: Hessian aware quantization of neural networks with mixed-precision. *Preprint*, arXiv:1905.03696.

Razvan-Gabriel Dumitru, Vikas Yadav, Rishabh Maheshwary, Paul-Ioan Clotan, Sathwik Tejaswi Madhusudhan, and Mihai Surdeanu. 2024. Layer-wise quantization: A pragmatic and effective method for quantizing llms beyond integer bit-levels. *Preprint*, arXiv:2406.17415.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. 2024. Extreme compression of large language models via additive quantization. *Preprint*, arXiv:2401.06118.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, and et.al. 2024. The language model evaluation harness.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and Aiesha Letman. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. 2024. Aptq: Attention-aware post-training mixed-precision quantization for large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, page 1–6. ACM.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and XIAOJUAN QI. 2024. BiLLM: Pushing the limit of post-training quantization for LLMs. In *Forty-first International Conference on Machine Learning*.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. 2025. Slim-llm: Salience-driven mixed-precision quantization for large language models. *Preprint*, arXiv:2405.14917.

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. *Preprint*, arXiv:1712.05877.

Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. 2023. Llm-mq: Mixed-precision quantization for efficient llm deployment. In *The Efficient Natural Language and Speech Processing Workshop with NeurIPS*, volume 9, page 3.

Zhen Li, Yupeng Su, Songmiao Wang, Runming Yang, Congkai Xie, Aofan Liu, Ming Li, Jiannong Cao, Yuan Xie, Ngai Wong, and Hongxia Yang. 2025. Infijanice: Joint analysis and in-situ correction engine for quantization-induced math degradation in large language models. *Preprint*, arXiv:2505.11574.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *MLSys*.

Lianbo Ma, Jianlun Ma, Yuee Zhou, Guoyang Xie, Qiang He, and Zhichao Lu. 2025. Learning from loss landscape: Generalizable mixed-precision quantization via adaptive sharpness-aware gradient aligning.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, and Gregory Chanan et.al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

QwenTeam. 2025. Qwen3.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Yuzhang Shang, Zhihang Yuan, Qiang Wu, and Zhen Dong. 2023. Pb-llm: Partially binarized large language models. *Preprint*, arXiv:2310.00034.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. *Preprint*, arXiv:2308.13137.

Zunhai Su, Qingyuan Li, Hao Zhang, YuLei Qian, Yuchen Xie, and Kehong Yuan. 2025. Unveiling super experts in mixture-of-experts large language models. *arXiv preprint arXiv:2507.23279*.

Zunhai Su and Kehong Yuan. 2025. Kvsink: Understanding and enhancing the preservation of attention sinks in kv cache quantization for llms. *arXiv preprint arXiv:2508.04257*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, and Nikolay Bashlykov. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *Preprint*, arXiv:2402.04396.

Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. 2025. Qtip: Quantization with trellises and incoherence processing. *Preprint*, arXiv:2406.11235.

Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. 2024. Diff-erank: A novel rank-based metric for evaluating large language models. *Preprint*, arXiv:2401.17139.

He Xiao, Runming Yang, Qingyao Yang, Wendong Xu, Zhen Li, Yupeng Su, Zhengwu Liu, Hongxia Yang, and Ngai Wong. 2025. Ptqtp: Post-training quantization to trit-planes for large language models. *Preprint*, arXiv:2509.16989.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, and Bo Zheng. 2024. Qwen2.5 Technical Report. *arXiv e-prints*, arXiv:2412.15115.

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. 2021. Hawqv3: Dyadic neural network quantization. *Preprint*, arXiv:2011.10680.

Mengxia Yu, De Wang, Qi Shan, Colorado J Reed, and Alvin Wan. 2024. The super weight in large language models. *arXiv preprint arXiv:2411.07191*.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. 2023. Rptq: Reorder-based post-training quantization for large language models. *Preprint*, arXiv:2304.01089.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

A More Result

As illustrated in Figure 6, we further shows the measurement for LLaMA3.X family models using LieQ and its diagnostic strategy.

LLM Usage Disclosure

This paper is primarily the work of the human authors, but we also made use of several advanced LLMs, including ChatGPT-5 and Kimi-K2. They were employed to support result analysis, and help with formatting and language polishing. We acknowledge the contributions of these LLMs, while fully recognizing their limitations, and take full responsibility for all content presented under our names. We did not include hidden prompt-injection text in the submission, and all external data and code comply with their respective licenses.

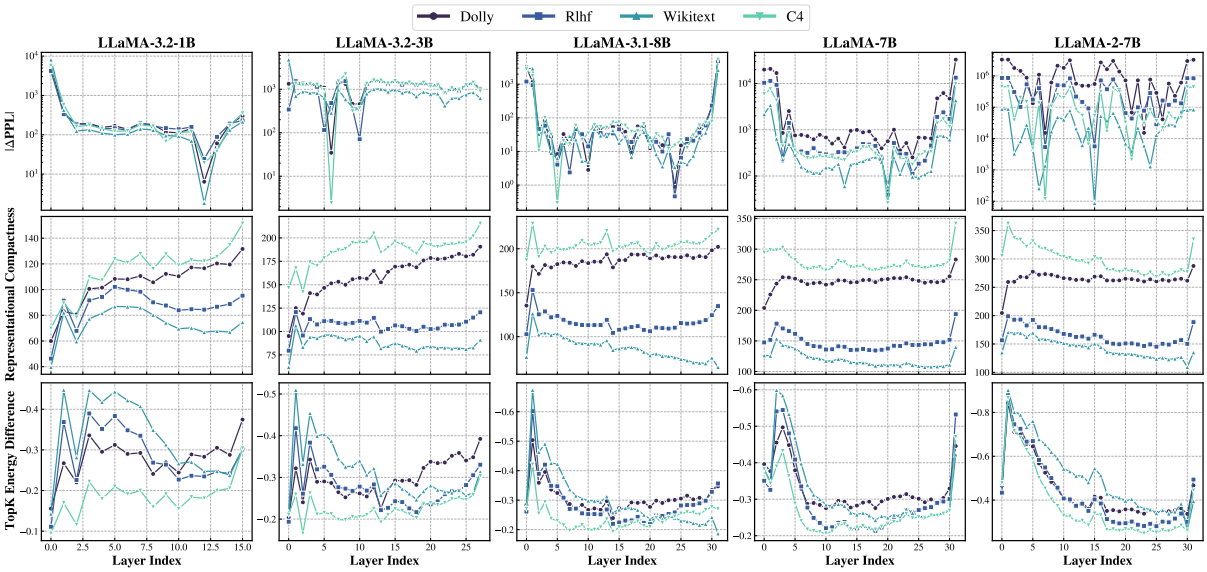

Figure 6: Functional diagnostic measures the drop in perplexity when a layer is removed on LLaMA3 family.