ASQ: Agent-guided Semantic-aware Quantization for Large Language Models

Seungdong Yoa[1] Ye Seul Sim[1] Suhee Yoon[1] Sanghyu Yoon[1] Dongmin Kim[1]
Soonyoung Lee[1] Junhyun Lee[2,3]* Bumsoo Kim[4]*

[1]LG AI Research [2]Hankuk University of Foreign Studies
[3]Noah's Farm [4]Chung-Ang University

{seungdong.yoa, ysl.sim, suhee.yoon, sanghyu.yoon, dmkim}@lgresearch.ai
soonyoung.lee@lgresearch.ai junhyun.lee@hufs.ac.kr bumsoo@cau.ac.kr

Abstract

Post-training quantization (PTQ) compresses large language models (LLMs) but remains largely semantic-agnostic, relying on global activation magnitude to guide precision allocation. This overlooks the functional heterogeneity of Transformer representations, where semantically meaningful channels are sparsely activated while structurally dominant channels exhibit large but context-insensitive activations. We propose Agent-guided Semantic-aware Quantization (ASQ), a PTQ framework that conditions on semantically informative positions identified by an auxiliary LLM. A dual-pathway scoring mechanism captures both activation selectivity and attention routing toward these positions, producing channel-wise importance scores for scaling. This yields a closer approximation to a semantically conditioned objective and consistently improves low-bit (e.g., W4A16) quantization performance across LLMs. Extensive experiments show consistent gains over state-of-the-art PTQ, preserving general reasoning and even surpassing FP16 performance on several benchmarks in specialized domains, while enabling training-free, privacy-preserving self-quantization with zero inference overhead.

1 Introduction

Large language models (LLMs) have achieved remarkable performance across a wide range of tasks, but their deployment remains challenging due to substantial memory and computational requirements. Post-training quantization (PTQ) has emerged as an effective approach for reducing model size and inference cost without retraining. Recent methods such as GPTQ [1], SmoothQuant [2], and AWQ [3] leverage activation statistics to guide weight quantization, significantly improving accuracy under low-bit settings.

Despite their success, existing PTQ methods are largely *semantic-agnostic*. They rely on global activation statistics—such as average magnitude—to determine which channels should be preserved during quantization. However, a growing body of work shows that Transformer representations are functionally heterogeneous: some neurons encode interpretable or factual patterns [4, 5], while others exhibit large, input-stable activations associated with structural or rescaling roles [6–9]. As a result, global activation magnitude is not a reliable proxy for semantic importance. Channels with high average magnitude may correspond to generic or structural behavior, while channels that encode semantically meaningful information may only be selectively activated in specific contexts.

A central challenge is that such semantically informative contexts are not directly identifiable from global statistics alone. Their signals are often sparse and diluted when aggregated across all positions

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

in the calibration data. Consequently, conventional PTQ methods may allocate precision to channels that are globally prominent but semantically uninformative, while under-protecting channels that are crucial for preserving meaningful content. This precision mismatch becomes exceptionally fatal in specialized domains (e.g., medical, legal, and financial), where preserving sparse but highly specialized vocabulary is critical to zero-shot reasoning.

In this work, we address this limitation by introducing *Agent-guided Semantic-aware Quantization* (ASQ). Instead of attempting to infer semantic importance purely from activation distributions, we leverage an auxiliary language model to identify semantically salient terms that already occur in the calibration text. We then locate their original token positions via exact matching, forming a subset of positions that serves as a conditioning signal. Importantly, this process does not modify or reorder the input sequence, nor does it perform token-level modeling. Rather, it provides a lightweight mechanism to probe how each channel responds under semantically informative contexts.

Building on this conditioning signal, we propose a dual-pathway semantic scoring mechanism. The first pathway measures *contrastive activation selectivity*, identifying channels whose activations are disproportionately elevated at the identified positions relative to their global baseline. This isolates channels that are selectively responsive to semantically meaningful content from those that are uniformly active. The second pathway captures *attention routing*, quantifying how strongly attention heads direct information toward these positions. Together, these signals characterize both where semantic information is encoded and how it is propagated through the network.

Finally, we incorporate the resulting semantic scores into activation-aware scaling for weight quantization. By reallocating precision toward channels that are selectively activated under semantically informative contexts, our method better preserves meaningful information under low-bit constraints, while remaining simple and compatible with existing PTQ pipelines.

Our contributions are summarized as follows:

- We identify a fundamental limitation of existing PTQ methods: global activation statistics do not reliably capture semantic importance due to the functional heterogeneity of Transformer representations.
- We introduce an agent-guided semantic conditioning strategy that identifies semantically salient positions in calibration data without modifying the input sequence or performing token-level modeling.
- We propose a dual-pathway scoring mechanism that captures both activation selectivity and attention-based routing under this conditioning signal.
- We develop a semantic-aware scaling rule that reallocates quantization precision toward channels selectively activated in semantically informative contexts, improving performance under low-bit quantization.

2 Preliminaries

We briefly revisit the activation-aware scaling framework [3], which forms the basis of our approach to weight quantization.

Consider a linear layer $Y = XW$, where $X \in \mathbb{R}^{T \times c_{\text{in}}}$ is the input activation and $W \in \mathbb{R}^{c_{\text{in}} \times c_{\text{out}}}$ is the weight matrix. A key observation is that quantization error depends on both the weight perturbation and the scale of input activations. To control this interaction, activation-aware methods apply a channel-wise rescaling transformation:

$$Y = \big(X \cdot \mathrm{diag}(\mathbf{s})^{-1}\big)\big(\mathrm{diag}(\mathbf{s}) \cdot W\big), \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^{c_{\text{in}}}$ is a positive scaling vector.

This transformation preserves the exact output of the layer, while redistributing magnitude between activations and weights. The rescaled weight

$$\widehat{W} = \mathrm{diag}(\mathbf{s}) \cdot W \tag{2}$$

is then quantized using uniform low-precision quantization (e.g., INT4). Because the inverse scaling $\mathrm{diag}(\mathbf{s})^{-1}$ can be absorbed into preceding operations (such as LayerNorm) during preprocessing, inference incurs no additional computational overhead.

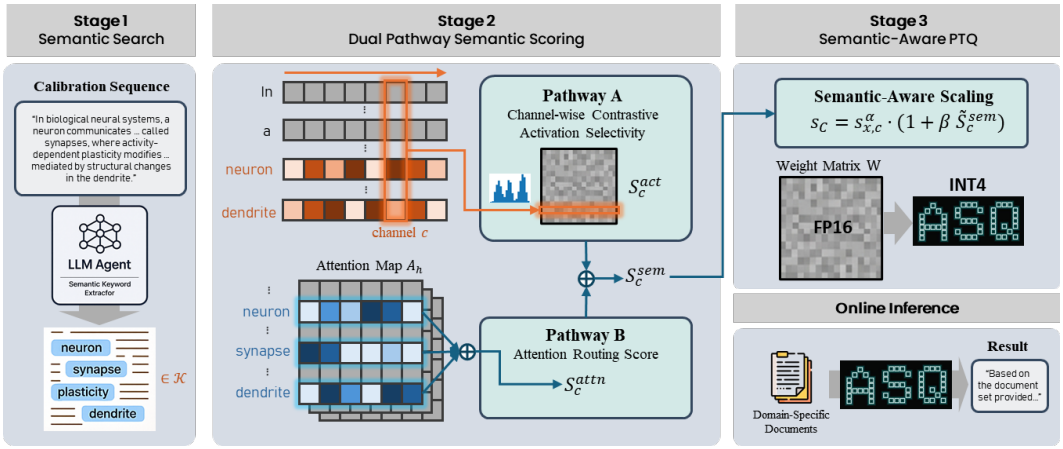


Figure 1: **Overview of our proposed method.** An auxiliary LLM agent identifies semantically salient positions in the calibration text, which condition dual-pathway scoring to produce channel-wise semantic importance scores for quantization scaling.

From a quantization perspective, this rescaling effectively controls the dynamic range of each input channel. Under uniform quantization, the rounding error magnitude is proportional to the quantization step size, which depends on the scale of $\widehat{W}$. Thus, increasing s_c reduces the relative quantization error associated with channel c, allowing more important channels to be protected.

In AWQ, the scaling vector is determined solely by activation magnitude:

$$s_c = s_{x,c}^{\alpha}, \quad s_{x,c} = \mathbb{E}[|X_{:,c}|], \tag{3}$$

where $\alpha \in [0, 1]$ controls the strength of protection. This strategy prioritizes channels with large average activation, but does not distinguish between channels that are uniformly active and those selectively engaged by semantically meaningful contexts. Addressing this limitation is the focus of our method.

3 Method

Post-training quantization (PTQ) approximates a pretrained weight matrix W with a low-precision version $\widetilde{W}$, introducing perturbation $\Delta W = \widetilde{W} - W$. As discussed in Section 2, $\widetilde{W}$ is the effective weight resulting from the quantization of the rescaled matrix $\widehat{W}$. For an input activation x, the induced output error is:

$$e(x) = x\Delta W. \tag{4}$$

The expected reconstruction error over a calibration distribution is:

$$\mathcal{L} = \mathbb{E}_x \left[\|x\Delta W\|_2^2 \right]. \tag{5}$$

Activation-aware PTQ methods reduce this error by assigning larger scaling factors to channels with high activation magnitude. However, such approaches are *semantic-agnostic*: they treat all channels with large activations equally, regardless of whether those activations correspond to meaningful content or generic structural behavior.

In large language models, this assumption is problematic. Transformer layers often contain structural outlier channels that exhibit consistently large activations across all positions, independent of context. At the same time, semantically meaningful features tend to be sparsely and selectively activated. As a result, global activation magnitude is not a reliable indicator of semantic importance.

To address this limitation, we propose a semantic-aware quantization framework that identifies channels selectively activated at semantically informative positions and allocates them higher precision during quantization.

3.1 Agent-Guided Semantic Conditioning

A key challenge is that semantically informative positions are not directly identifiable from global activation statistics. To address this, we introduce a lightweight mechanism to locate positions likely to carry meaningful content.

Given a calibration sequence of length T, we prompt an auxiliary language model to extract semantically salient terms that explicitly appear in the text. These terms are mapped to their corresponding token indices via exact string matching, yielding a subset:

$$\mathcal{K} \subset \{1, \ldots, T\}. \tag{6}$$

Importantly, $\mathcal{K}$ is used only to restrict activation and attention statistics to a semantically informative subset of positions. We do not modify or reweight tokens, nor do we perform token-level modeling. This allows us to probe how each channel behaves under semantically relevant contexts without altering the input sequence.

3.2 Dual-Pathway Semantic Scoring

We estimate the semantic importance of each channel using two complementary signals that capture activation selectivity and attention routing.

Pathway A: Contrastive Activation Selectivity for Semantic Channels. A naïve reliance on activation magnitude is insufficient due to the presence of structural outliers—channels that exhibit consistently large values across all positions regardless of context. To isolate channels that are selectively responsive to semantically meaningful content, we compute, for each layer l and input channel c:

$$S_c^{\text{act}} = \frac{\mathbb{E}[|X_{\mathcal{K},c}|]}{\mathbb{E}[|X_{:,c}|] + \epsilon}, \tag{7}$$

where c denotes the input channel index. $X_{\mathcal{K},c}$ represents the activations of channel c at the agent-identified salient positions $\mathcal{K}$, and $X_{:,c}$ denotes its activations over all token positions in the sequence. Because $\mathcal{K}$ typically represents a sparse subset of the sequence, $\mathbb{E}[|X_{:,c}|]$ serves as a stable baseline for each channel. Channels that are uniformly active receive scores close to one, while channels that exhibit disproportionately large activations at positions in $\mathcal{K}$ receive higher scores. This contrastive formulation suppresses globally large but uninformative channels and highlights those selectively activated by semantically meaningful content.

Pathway B: Attention Routing for Capturing Semantic Propagation. While Pathway A captures which channels are selectively activated at semantic positions, it does not account for how information at those positions is propagated to the rest of the sequence. In Transformer architectures, attention heads control this propagation: a head that directs large attention mass toward positions in $\mathcal{K}$ causes their value representations to be aggregated into outputs across the sequence, amplifying the influence of semantically marked content.

For each attention head h in each layer, we compute the routing score S_h^{attn} by measuring the average attention mass directed toward positions in $\mathcal{K}$:

$$S_h^{\text{attn}} = \frac{1}{T \cdot |\mathcal{K}|} \sum_{t=1}^{T} \sum_{k \in \mathcal{K}} A_h(t, k), \tag{8}$$

where $A_h(t, k)$ is the attention probability from query position t to key position k. This score measures the average attention probability that any query position assigns to positions in $\mathcal{K}$. A higher score indicates that the head routes more value information from semantically marked positions into outputs across the sequence.

To incorporate this head-level signal into channel-wise importance, we map each head score to the input channels of the output projection W_O:

$$S_c^{\text{attn}} = S_{\lfloor c/d_h \rfloor}^{\text{attn}}, \tag{9}$$

where d_h denotes the dimension per head. Since the input to W_O is the concatenation of all head outputs, channel c corresponds to head $h = \lfloor c/d_h \rfloor$. This mapping assigns a uniform score across all

d_h channels within a head, reflecting that the head-level routing signal does not distinguish among individual dimensions within a head.

To obtain the final semantic importance, we combine the scores from both pathways. Since the two signals operate on different scales, we rescale the attention-based scores S_c^{attn} to match the maximum magnitude of the activation-based scores S_c^{act}:

$$\hat{S}_c^{\text{attn}} = S_c^{\text{attn}} \cdot \frac{\max_{c'} S_{c'}^{\text{act}}}{\max_{c'} S_{c'}^{\text{attn}} + \epsilon}, \tag{10}$$

where ϵ is a small constant for numerical stability. The final combined semantic importance score is defined as:

$$S_c^{\text{sem}} = S_c^{\text{act}} + \hat{S}_c^{\text{attn}}. \tag{11}$$

This formulation captures both the selective activation of channels and the routing of information toward semantically informative positions, ensuring both pathways contribute on a comparable scale.

Notably, this dual-pathway combination is applied specifically to output projection (W_O) layers where the head-to-channel correspondence is explicitly defined by the concatenated-head structure. For all other linear layers (e.g., FFN or Q, K, V projections), we set $S_c^{\text{sem}} = S_c^{\text{act}}$.

3.3 Semantic-Aware Scaling

We incorporate the semantic scores into quantization by modulating channel-wise scaling factors. Under uniform quantization, the magnitude of weight perturbation for a specific channel is inversely related to the applied scaling:

$$\|\Delta W_{c,:}\|_2^2 \propto \frac{1}{s_c^2}. \tag{12}$$

Conventional activation-aware methods use $s_c = s_{x,c}^{\alpha}$, where $s_{x,c} = \mathbb{E}[|X_{:,c}|]$ is the mean absolute activation. We instead augment this with a semantic-dependent factor:

$$s_c = s_{x,c}^{\alpha} \cdot \left(1 + \beta \tilde{S}_c^{\text{sem}}\right), \tag{13}$$

where $\tilde{S}_c^{\text{sem}}$ is obtained by applying per-layer min-max normalization to S_c^{sem}, i.e., $\tilde{S}_c^{\text{sem}} = (S_c^{\text{sem}} - \min_{c'} S_{c'}^{\text{sem}})/(\max_{c'} S_{c'}^{\text{sem}} - \min_{c'} S_{c'}^{\text{sem}} + \epsilon)$, so that $\tilde{S}_c^{\text{sem}} \in [0, 1]$. Since the raw scale of S_c^{sem} varies across layers and differs from that of $s_{x,c}^{\alpha}$, this normalization ensures that the boost factor $(1 + \beta \tilde{S}_c^{\text{sem}})$ ranges consistently from 1 to $1 + \beta$, making β interpretable as the maximum relative increase in scaling for the most semantically selective channel. This increases the effective precision of channels that are selectively activated at positions in $\mathcal{K}$, thereby reducing their contribution to quantization error. Channels without such selectivity revert to the baseline scaling.

This design is particularly effective in large language models, where semantic features are sparsely activated, structural channels dominate global magnitude, and attention selectively routes information across positions. By aligning scaling with selective activation rather than global magnitude, the proposed method more effectively preserves semantically meaningful information under low-bit quantization.

4 Theoretical Analysis

For a linear layer with weight perturbation ΔW, the reconstruction error under a diagonal approximation is:

$$\mathcal{L} = \mathbb{E}_x \left[\|x\Delta W\|_2^2\right] \approx \sum_c \|\Delta W_{c,:}\|_2^2 \cdot \mathbb{E}[x_c^2], \tag{14}$$

which motivates magnitude-based scaling using global activation statistics.

However, in language models, task-relevant information is concentrated in a subset of positions $\mathcal{K}$. A more appropriate objective is:

$$\mathcal{L}^{\text{sem}} \approx \sum_c \|\Delta W_{c,:}\|_2^2 \cdot \mathbb{E}[x_c^2 \mid t \in \mathcal{K}]. \tag{15}$$

Our contrastive score

$$S_c^{\text{sem}} \approx \frac{\mathbb{E}[|x_c| \mid t \in \mathcal{K}]}{\mathbb{E}[|x_c|]} \tag{16}$$

highlights channels selectively activated at semantically informative positions.

Using the approximation $\|\Delta W_{c,:}\|_2^2 \propto 1/s_c^2$, we obtain:

$$\mathcal{L}^{\text{sem}} \approx \sum_c \frac{w_c^{\text{sem}}}{s_c^2}, \tag{17}$$

where $w_c^{\text{sem}} = \mathbb{E}[x_c^2 \mid t \in \mathcal{K}]$ denotes the semantic importance weight of channel c. Under a fixed scale normalization (e.g., $\prod_c s_c = 1$), minimizing this approximate objective yields the relative optimum $s_c \propto \sqrt{w_c^{\text{sem}}}$.

Our scaling rule

$$s_c = s_{x,c}^{\alpha} \cdot (1 + \beta \tilde{S}_c^{\text{sem}}) \tag{18}$$

combines global magnitude with semantic selectivity, providing a practical approximation motivated by this relative allocation when semantic importance is not aligned with global activation—a common regime in large language models, where magnitude-based scaling misallocates precision while semantic-aware scaling better preserves task-relevant information.

5 Experiments

5.1 Experimental Setup

Models and Baselines. We evaluate on two open-weight LLMs spanning different scales: Meta-Llama-3-8B [10] and Qwen2.5-32B [11]. All quantization baselines use 4-bit weight-only symmetric min-max quantization with a group size of 128. We compare ASQ against full-precision (FP16) upper bounds, naive Round-To-Nearest (RTN), GPTQ [1] with Hessian-based weight rounding, and AWQ [3] with activation-magnitude scaling.

Calibration Data and Zero-Shot Transfer Rigor. We focus on specialized domains, where preserving sparse but critical terminology is essential for reliable zero-shot reasoning—a regime where such terms are easily overshadowed by global activation statistics. For the domain-specific evaluation, we align calibration data with the target domain while strictly separating calibration and evaluation data to rigorously test zero-shot transfer. Crucially, no QA-formatted data is exposed during calibration; the model is quantized solely on domain-specific raw text and evaluated on unseen QA tasks. Calibration uses raw text from PMC-Patients V2 [12] (Medical), Pile of Law [13] (Legal), and SEC Filings-10K (EDGAR-CORPUS) [14] (Finance), with 128 samples per domain (max length 1,024 tokens). Preprocessing details are in Appendix A.

Evaluation Benchmarks. We evaluate zero-shot multiple-choice accuracy on PubMedQA [15], MedQA-USMLE [16], and MedMCQA [17] for Medical; LexGLUE CaseHOLD [18] for Legal; and MMLU-Economics (FinanceQA) [19] for Finance. None of these benchmarks are seen during calibration.

Implementation Details. All quantized models are simulated in FP16 (pseudo-quantization) under the W4A16 configuration. We employ Gemini 2.5 Flash [20] as the auxiliary LLM agent for keyword extraction from the calibration corpora; S_c^{sem} is then computed by measuring how strongly the target model's channels activate at these keywords versus generic tokens. Full hyperparameter search ranges and other settings are in Appendix D.

5.2 Main Results: Zero-Shot Domain Preservation

Table 1 presents zero-shot accuracy across five specialized QA benchmarks under W4A16 quantization, rigorously testing whether quantization preserves latent domain knowledge rather than surface-level pattern matching.

ASQ consistently achieves the best accuracy among all 4-bit methods. On Llama-3-8B, ASQ attains an average of 42.51%, recovering 99.0% of the FP16 baseline (42.95%) and outperforming the next-best quantized method, RTN, by +1.96 pp. The gap is larger relative to AWQ (+2.30 pp)

Table 1: Zero-shot accuracy (%) comparison across specialized domains. **Bold** indicates the best performance among 4-bit quantized models. ASQ consistently outperforms existing post-training quantization (PTQ) baselines. Notably, ASQ even surpasses the FP16 baseline in several benchmarks (e.g., PubMedQA, LexGLUE), highlighting its robust semantic preservation.

Method	Medical			Legal	Finance	Avg.
	MedQA	MedMCQA	PubMedQA	LexGLUE	FinanceQA	
Llama-3-8B (W4A16)						
FP16 (Dense)	42.81	36.94	62.80	25.58	46.63	42.95
RTN	39.25	37.09	58.70	26.58	41.11	40.55
GPTQ	34.01	34.14	54.80	26.19	34.91	36.81
AWQ	36.53	35.52	61.80	26.63	40.57	40.21
ASQ (Ours)	**39.98**	**37.32**	**65.40**	**26.72**	**43.13**	**42.51**
Qwen2.5-32B (W4A16)						
FP16 (Dense)	48.15	38.78	60.90	28.11	53.77	45.94
RTN	46.82	38.18	60.40	28.33	51.62	45.07
GPTQ	43.52	36.53	56.20	25.94	53.23	43.09
AWQ	47.21	38.23	61.70	28.14	52.29	45.51
ASQ (Ours)	**47.60**	**38.97**	**62.50**	**28.72**	**53.43**	**46.24**

and GPTQ (+5.70 pp), indicating that activation-magnitude scaling alone is insufficient (notably, AWQ underperforms even RTN: 40.21% vs. 40.55%) and that the contrastive semantic signal in ASQ provides a complementary and crucial advantage. On the larger Qwen2.5-32B, ASQ achieves 46.24%, which reaches parity with and marginally exceeds the FP16 baseline of 45.94%. This suggests that well-targeted salient-channel protection can act as an implicit regularizer at scale, potentially suppressing noise in less important channels while preserving semantically critical ones.

ASQ surpasses FP16 on individual benchmarks. Notably, a 4-bit quantized model can exceed its full-precision counterpart on several domain benchmarks. On Llama-3-8B, ASQ outperforms FP16 on PubMedQA (65.40% vs. 62.80%, +2.60 pp), MedMCQA (37.32% vs. 36.94%), and LexGLUE (26.72% vs. 25.58%). This pattern is consistent on Qwen2.5-32B, where ASQ again exceeds FP16 on PubMedQA (+1.60 pp), MedMCQA (+0.19 pp), and LexGLUE (+0.61 pp). We attribute this to the noise-suppression effect of quantization on low-importance channels: by concentrating representational capacity on semantically salient channels identified through contrastive scoring, ASQ effectively denoises the weight space, yielding sharper domain-relevant representations.

6 Analysis

6.1 Magnitude vs. Semantic: Channel Selection Divergence

Our semantic-aware scaling (Eq. 13) combines two factors: the activation magnitude $\mathbb{E}[|X_{:,c}|]^{\alpha}$, which forms the base protection shared with AWQ, and a contrastive semantic boost $1 + \beta \cdot \tilde{S}_c^{\text{sem}}$. A natural question is whether the semantic signal identifies channels that the magnitude signal already protects—in which case the boost would be redundant—or whether it captures a genuinely complementary set.

Table 2 answers this decisively. We select the top-5% channels under each signal and measure their Jaccard overlap. Across both models and all domains, the overlap is low (avg. Jaccard ≈ 0.055), meaning that approximately 90% of the channels our contrastive score prioritizes are not selected by magnitude-based ranking.

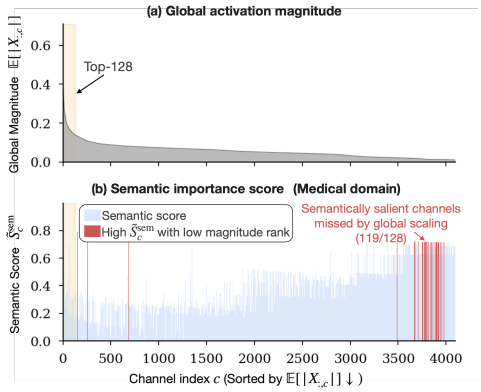


Figure 2: Global magnitude vs. semantic importance. Red bars mark the 119/128 salient channels missed by magnitude ranking.

This orthogonality is most extreme at the output projection (`o_proj`), where Jaccard drops to 0.000. At this layer, S_c^{sem} combines contrastive activation selectivity (Pathway A) with head-level attention routing toward domain keywords (Pathway B); the latter introduces a reweighting that is structurally

Table 2: Channel overlap between activation magnitude ($\mathbb{E}[|X_{:,c}|]$, AWQ) and our contrastive score (S_c^{sem}). We report Jaccard similarity of the top-5% channels selected by each signal. Near-zero overlap indicates that our method identifies an almost entirely disjoint set of important channels.

Model	Layer Type	Med	Leg	Fin	Avg.
Llama-3-8B	$W_Q/W_K/W_V$	0.108	0.044	0.060	0.071
	W_O	0.000	0.000	0.000	0.000
	FFN	0.102	0.032	0.045	0.060
	All	**0.090**	**0.032**	**0.045**	**0.056**
Qwen2.5-32B	$W_Q/W_K/W_V$	0.071	0.055	0.075	0.067
	W_O	0.000	0.000	0.002	0.001
	FFN	0.084	0.044	0.054	0.061
	All	**0.066**	**0.043**	**0.056**	**0.055**

Table 3: Semantic quantization error analysis on Llama-3-8B (Medical), normalized to RTN = 1.000. Lower is better.

Method	Top-5% Semantic Channel Error (↓)	Semantically Weighted Error ($\mathcal{L}_{\text{sem}}$, ↓)
RTN	1.000	1.000
AWQ	1.180	1.067
ASQ (Ours)	**0.987**	**0.959**

decoupled from per-channel activation magnitude, driving the selected channels far from the magnitude top set. Even for q/k/v projections and FFN layers, where only Pathway A applies, the contrastive normalization $S_c^{\text{sem}} = \mathbb{E}[|X_{\mathcal{K},c}|] \,/\, (\mathbb{E}[|X_{:,c}|] + \epsilon)$ effectively removes structurally large channels from the top set, surfacing small-magnitude but domain-responsive "expert neurons" that magnitude alone would never protect.

Figure 2 provides a visual confirmation of this phenomenon for Llama-3-8B (Layer 15, Medical): 119 out of the top-128 channels prioritized by our semantic score $\tilde{S}_c^{\text{sem}}$ (vs. the top-128 by magnitude) are hidden in the low-magnitude tail of $\mathbb{E}[|X_{:,c}|]$, demonstrating that global scaling systematically neglects these domain-critical neurons.

This result explains why the semantic boost in Eq. 13 yields consistent gains over AWQ: it allocates quantization precision to a complementary set of channels that carry domain-specific knowledge but lack the structural prominence needed to survive magnitude-based selection.

6.2 Understanding the Domain-Specific Baseline Reversal

In Table 1, AWQ underperforms RTN on Llama-3-8B (40.21% vs. 40.55%), despite its expected advantage under general calibration. As shown in Section 6.1, magnitude-based scaling can overlook semantically important channels. To understand this reversal, we directly analyze the quantization error on semantic channels.

We compute the effective weight perturbation and the corresponding per-input-channel squared error:

$$\Delta W_{\text{eff}} = \text{diag}(s)^{-1} Q(\text{diag}(s) W) - W, \quad \text{err}_c = \|\Delta W_{\text{eff},c,:}\|_2^2.$$

We then evaluate (i) the error on the top-5% channels ranked by our semantic importance score and (ii) the semantically weighted reconstruction error $\mathcal{L}^{\text{sem}}$ in Eq. 15, using err_c and the conditional second moments $\mathbb{E}[x_c^2 \mid t \in \mathcal{K}]$. Both the semantic channel rankings and conditional second moments are computed once from FP16 activations at agent-identified positions and held fixed across methods. Both metrics are aggregated across quantized linear layers.

As shown in Table 3, AWQ increases the measured error on the top-5% semantic channels by 18.0% over RTN and increases the semantically weighted error by 6.7%. In contrast, ASQ restores the top semantic-channel error to approximately the RTN level and reduces the semantic reconstruction objective by 4.1%. This is consistent with the mechanism that AWQ's magnitude-based scaling can under-protect low-magnitude semantic channels (Figure 2), whereas RTN treats channels uniformly and avoids selectively increasing their error.

Table 4: Cross-domain channel overlap analysis. We measure the Jaccard similarity of the top-15% important channels between domain pairs under three importance signals. Lower overlap indicates stronger domain specificity.

Model	Importance Signal	Med–Leg	Med–Fin	Leg–Fin	Avg.
Llama-3-8B	$\mathbb{E}[\lvert X_{:,c}\rvert]$ (magnitude, AWQ)	0.550	0.533	0.599	0.561
	$\mathbb{E}[\lvert X_{\mathcal{K},c}\rvert]$ (keyword activation)	0.325	0.419	0.380	0.375
	S_c^{sem} (contrastive, ours)	**0.095**	**0.109**	**0.130**	**0.111**
Qwen2.5-32B	$\mathbb{E}[\lvert X_{:,c}\rvert]$ (magnitude, AWQ)	0.622	0.612	0.666	0.633
	$\mathbb{E}[\lvert X_{\mathcal{K},c}\rvert]$ (keyword activation)	0.346	0.385	0.430	0.387
	S_c^{sem} (contrastive, ours)	**0.127**	**0.137**	**0.204**	**0.156**

Table 5: General reasoning preservation under domain-specific calibration (W4A16). All methods are calibrated with domain text and evaluated on domain-agnostic benchmarks.

Model	Method	MMLU-G	ARC-E	ARC-C	Avg.
Llama-3-8B	FP16 (Dense)	47.14	76.62	53.39	59.05
	RTN	44.65	74.80	50.39	56.61
	AWQ	45.30	75.36	50.44	57.03
	ASQ (Ours)	46.21	75.26	50.65	57.37
Qwen2.5-32B	FP16 (Dense)	55.10	75.22	54.68	61.67
	RTN	53.77	73.95	53.39	60.37
	AWQ	54.08	74.20	53.44	60.57
	ASQ (Ours)	54.53	74.31	53.62	60.82

Interestingly, this ordering is not universal. Under general-text calibration (C4) and evaluation on five general benchmarks, AWQ and GPTQ outperform RTN on every task, while ASQ achieves the best accuracy among all quantized methods (Appendix G).

6.3 Cross-Domain Channel Specialization

A key assumption of our method is that different domains activate distinct "expert" channels within the network. To verify this, we measure the Jaccard similarity of the top-15% important channels across all domain pairs under three importance signals (Table 4).

The global activation magnitude $\mathbb{E}[|X_{:,c}|]$ used by AWQ exhibits high cross-domain overlap (avg. 0.56–0.63), indicating that magnitude-based importance is largely domain-agnostic—it reflects structural channel properties rather than semantic specialization. Using keyword-position activations $\mathbb{E}[|X_{\mathcal{K},c}|]$ alone reduces avg. overlap to 0.38–0.39, but this unnormalized signal still conflates genuinely specialized channels with those that are simply large everywhere, including at keyword positions.

Our contrastive score S_c^{sem} resolves this by normalizing keyword activation against the channel's overall baseline. The cross-domain overlap drops sharply to 0.11–0.16—meaning that approximately 74–81% of the selected channels are not shared between domain pairs. This 2.5–3.4$\times$ reduction compared to the unnormalized signal confirms that the contrastive formulation effectively isolates domain-expert neurons from structural outliers, validating the need for semantic-aware channel selection beyond magnitude-based approaches. The trend is consistent across Llama-3-8B and Qwen2.5-32B, suggesting that this specialization is not limited to a single model architecture.

6.4 Preservation of General Reasoning

A potential concern with domain-specific calibration is that over-protecting domain channels may sacrifice the model's general reasoning capability. To investigate this, we quantize each model using domain-specific calibration (as in our main experiments) but evaluate on general-knowledge benchmarks unrelated to any calibration domain: MMLU general subjects (e.g., philosophy, astronomy, computer science) and ARC (Easy and Challenge). Results are averaged across three calibration domains.

Table 5 shows that ASQ retains general reasoning performance on par with or slightly above AWQ, despite its domain-focused calibration. The semantic boost selectively protects domain-expert channels *in addition to* the magnitude-based baseline, rather than re-allocating precision away from general-purpose channels. This confirms that domain-aware quantization is not a zero-sum trade-off: specialized knowledge can be preserved without degrading general capabilities.

6.5 Self-Quantization: The Model as Its Own Agent

Our framework requires an agent to extract domain keywords from calibration text. While an external LLM (Gemini-2.5-Flash) serves this role in our main experiments, we investigate whether the target model itself can act as its own agent.

Table 6: Effect of agent choice (W4A16). *External*: Gemini-2.5-Flash; *Self*: the target model (FP16).

Model	Agent	Acc. (%)
Llama-3-8B	External (Gemini)	42.51
	Self (FP16)	41.73
Qwen2.5-32B	External (Gemini)	46.24
	Self (FP16)	45.88

Table 6 shows that self-quantization—using the FP16 target model for keyword extraction—incurs only 0.4–0.8%p accuracy degradation compared to the external agent, while still improving over vanilla AWQ by 0.4–1.5%p. The gap between self and external modes is smaller for the larger model (0.36%p for Qwen2.5-32B vs. 0.78%p for Llama-3-8B), suggesting that larger models may better serve as their own agents.

This result has two implications. First, it eliminates the dependence on external APIs, enabling fully offline, cost-free deployment. Second, it suggests that a pretrained LLM already knows where its specialized knowledge resides—the same internal representations that encode domain expertise are sufficiently accessible for the model to guide its own compression.

6.6 Ablation Studies

To isolate the contributions of the two pathways, we evaluate activation selectivity (A), attention routing (B), and their combination under the Llama-3-8B setting in Table 1. All variants retain AWQ's magnitude-based scaling. As shown in Table 7, A-only and B-only improve over AWQ by 1.23 and 1.46 pp on average, respectively, while combining them yields a further 1.07 pp over A-only and 0.84 pp over B-only. These results confirm their complementary contributions.

Table 7: Pathway ablation on Llama-3-8B (W4A16).

Method	Avg. Acc. (%)
AWQ (no semantic term)	40.21
Pathway A only	41.44
Pathway B only	41.67
A+B (ASQ, ours)	**42.51**

Additional keyword-selection ablations show that coverage-matched random positions fail to improve over AWQ (40.06% vs. 40.21%), while the agent achieves higher accuracy than dictionary and NER alternatives (Appendix H). ASQ introduces no additional inference-time operators relative to AWQ; its offline calibration overhead and memory usage are analyzed in Appendix I.

7 Conclusion

We presented ASQ, a post-training quantization framework that leverages an auxiliary language model to identify and protect semantically salient channels during quantization. By combining contrastive activation selectivity with attention routing, ASQ separates channels encoding meaningful content from structurally dominant but semantically uninformative outliers—a distinction invisible to conventional magnitude-based methods. Experiments across two model scales and three specialized domains show that ASQ consistently outperforms existing PTQ baselines under W4A16, recovering 99.0% of FP16 accuracy on Llama-3-8B and surpassing it on Qwen2.5-32B. These results confirm that semantic-aware precision allocation is not a zero-sum trade-off: targeted channel protection preserves domain knowledge without degrading general capabilities.

Acknowledgments and Disclosure of Funding

This work was supported by the National IT Industry Promotion Agency (NIPA) grant funded by the Korea government (MSIT) (Development of Physical Intelligence Action Models and an Integrated Framework, RS-2026-25621677), the Institute of Information & communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development (IITP-2026-RS-2026-25545512) grant funded by the Korea government (MSIT), the Hankuk University of Foreign Studies Research Fund (2026) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25604335).

References

[1] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *ICLR*, 2023.

[2] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

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

[4] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.

[5] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *ACL*, 2022.

[6] Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. Bert busters: Outlier dimensions that disrupt transformers. In *Findings of ACL*, 2021.

[7] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *NeurIPS*, 2023.

[8] Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. In *NeurIPS*, 2024.

[9] Zihan Qiu, Zeyu Huang, Kaiyue Wen, et al. A unified view of attention and residual sinks: Outlier-driven rescaling is essential for transformer training. *arXiv preprint arXiv:2601.22966*, 2026.

[10] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[11] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

[12] Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Turun Peng, and Sheng Yu. PMC-Patients: A large-scale dataset of patient summaries and relations for benchmarking retrieval-based clinical decision support systems. *arXiv preprint arXiv:2202.13876*, 2023.

[13] Peter Henderson, Mark S Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel E Ho. Pile of law: Learning responsible data filtering from the law and a 256 GB open-source legal dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. EDGAR-CORPUS: Billions of tokens make the world go round. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, 2021.

[15] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[16] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[17] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning (CHIL)*, 2022.

[18] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[19] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

[20] Google DeepMind. Gemini 2.5: Our most intelligent AI model, 2025. Technical report. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`.

[21] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[22] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[24] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, 2020.

[26] Ye Lin, Yanyang Li, Tengbo Liu, Tong Xiao, Tongran Liu, and Jingbo Zhu. Towards fully 8-bit integer inference for the transformer model. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[27] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

[29] Changhai Zhou, Shiyang Zhang, Yuhua Zhou, Zekai Liu, and Shichao Weng. AutoMixQ: Self-adjusting quantization for high performance memory-efficient fine-tuning. *arXiv preprint arXiv:2411.13814*, 2025.

[30] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*, 2024.

[31] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[32] Lutfi Eren Erdogan, Nicholas Lee, Siddharth Jha, Sehoon Kim, Ryan Tabrizi, Suhong Moon, Coleman Hooper, Gopala Ananthanarayanan, Kurt Keutzer, and Amir Gholami. TinyAgent: Function calling at the edge. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 2024.

[33] Kaiyuan Deng, Hangyu Zheng, Minghai Qing, Kunxiong Zhu, Gen Li, Yang Xiao, Lan Emily Zhang, Linke Guo, Bo Hui, Yanzhi Wang, Geng Yuan, Gagan Agrawal, Wei Niu, and Xiaolong Ma. From bits to chips: An LLM-based hardware-aware quantization agent for streamlined deployment of LLMs. *arXiv preprint arXiv:2601.03484*, 2026.

A Calibration Data Preprocessing

To ensure the high quality and density of domain-specific signals during calibration, we apply strict filtering and preprocessing to the raw text corpora:

- **Medical:** We utilize raw patient case narratives from PMC-Patients V2 [12], which densely contain diagnostic and prognostic terminology without QA formatting.
- **Legal:** For the Pile of Law (CourtListener Opinions) [13], we skip the first 1,000 tokens of each document to remove uninformative boilerplate captions. We then filter for documents that retain at least 512 tokens of substantive legal analysis.
- **Finance:** From the EDGAR-CORPUS [14], we specifically target Item 7 (MD&A) and Item 1A (Risk Factors) of 10-K filings, keeping only meaningful sections over 500 characters to prevent the inclusion of sparse reference stubs.

B Evaluation Benchmark Details

To ensure full reproducibility of our zero-shot evaluation, we detail the specifications of the five specialized benchmarks used in this study. We evaluate all quantized models in a strict zero-shot setting using log-likelihood ranking. Specifically, for each candidate answer, we compute the mean per-token negative log-likelihood of the answer tokens conditioned on the question prompt, selecting the option with the lowest loss. No few-shot exemplars are provided. The maximum sequence length is set to 2,048 tokens for all benchmarks to prevent prompt truncation.

Table 8 summarizes the dataset statistics, and the domain-specific details are as follows:

Table 8: Summary of evaluation benchmarks, including source identifiers, splits, and task formats.

Benchmark	Hugging Face Source	Domain	Split	# Examples	# Choices
MedQA	`GBaker/MedQA-USMLE-4-options`	Medical	test	1,273	4
MedMCQA	`openlifescienceai/medmcqa`	Medical	val.	4,183	4
PubMedQA	`qiaojin/PubMedQA` (labeled)	Medical	all*	1,000	3
LexGLUE CaseHOLD	`coastalcph/lex_glue`	Legal	test	3,600	5
FinanceQA	`cais/mmlu` (3 econ. subsets)	Finance	test	742	4

*The `pqa_labeled` subset has no official split; all 1,000 expert-labeled samples are used.

Medical Domain.

- **MedQA (USMLE):** We use the 4-option variant containing clinical vignette-style questions from the United States Medical Licensing Examination. We report accuracy on the test split (1,273 examples).
- **MedMCQA:** Sourced from Indian medical entrance examinations (AIIMS and NEET PG). Because the official test-set labels are withheld, we evaluate on the validation split (4,183 examples).
- **PubMedQA:** A biomedical QA dataset paired with PubMed abstracts. The task is to answer yes, no, or maybe (3-way). The `pqa_labeled` subset consists of 1,000 expert-annotated examples without an official train/test split; we use all 1,000 examples for evaluation. Context is truncated to 800 characters to fit within the sequence budget.

Legal Domain.

- **LexGLUE CaseHOLD:** A legal holding classification task from LexGLUE. Given a legal context citing a particular case, the model must select the correct holding statement from five candidates. We evaluate on the test split (3,600 examples).

Finance Domain.

- **FinanceQA (MMLU Economics):** To assess financial and economic reasoning, we aggregate three economics-related subsets from the MMLU benchmark: *Econometrics*, *High*

School Macroeconomics, and *High School Microeconomics*. All evaluation is conducted on the respective test splits, totaling 742 examples.

C Agent Prompt Template

We employ Gemini 2.5 Flash [20] as the auxiliary LLM agent for semantic keyword extraction. The prompt template is intentionally minimal to avoid biasing the extraction toward predefined term categories:

Keyword Extraction Prompt

```
You are a {domain} expert.  Extract the most important
{domain}-specific terms that actually appear in the text below.  Quote
them exactly as they appear.  Output only the terms, comma-separated.
Text:  {calibration_text}

Terms:
```

where `{domain}` $\in$ {*medical*, *legal*, *finance*}.

The extracted terms are then mapped back to their original token positions via exact string matching (case-insensitive), forming the semantic subset $\mathcal{K}$. Terms shorter than 3 characters are discarded to prevent spurious matches with common substrings.

We deliberately avoid providing in-context examples or domain-specific term taxonomies, as doing so would introduce selection bias toward predefined categories. Instead, we rely on the agent's intrinsic linguistic knowledge to identify contextually salient terms, ensuring that the extraction generalizes across diverse texts within each domain. Note that the agent is not required to perform domain reasoning; it only needs to recognize domain-specific vocabulary that explicitly appears in the input, a capability well within the reach of modern instruction-tuned LLMs.

D Hyperparameter Configuration

Scale Optimization for ASQ. The scaling factor (Eq. 13) is determined by two hyperparameters: α (base activation exponent) and β (semantic boost strength). Transformer layers are evenly divided into G=8 groups, and each group independently selects (α, β) to minimize reconstruction MSE on the calibration set. Based on preliminary experiments, we constrain the search to $\alpha \in [0.1, 0.3]$ and $\beta \in [0.0, 0.5]$. Across all models and domains, the selected values typically fall in the range $\alpha \approx 0.20$–0.28 and $\beta \approx 0.05$–0.20, indicating that conservative base scaling combined with moderate semantic boosting is consistently preferred.

Comparison with AWQ's default. While the AWQ paper [3] reports a fixed α=0.5, its official implementation performs grid search over α. To ensure a fair comparison, we follow the same practice and search $\alpha \in [0.1, 0.5]$, selecting the value that minimizes calibration loss.

AWQ search-range robustness. We additionally verified that extending AWQ's search to the full 20-point grid $\alpha \in \{0, 0.05, \ldots, 0.95\}$, including the $\alpha = 0$ RTN fallback, yields nearly identical performance in our domain-specific setting (40.18% vs. 40.21% average accuracy).

Other settings.

- Calibration samples: 128 per domain (fixed seed 42)
- Calibration sequence length: 1,024 tokens
- Quantization: W4A16, symmetric, group size 128
- Minimum keyword length: 3 characters
- Hardware: 4$\times$ NVIDIA RTX PRO 6000 (96GB) GPUs

E Perplexity Analysis

Table 9 reports held-out perplexity on domain-specific calibration corpora for Llama-3-8B under W4A16 quantization. ASQ achieves perplexity comparable to RTN and AWQ across all three domains, confirming that semantic-aware scaling does not introduce additional language modeling degradation. GPTQ exhibits notably higher perplexity, particularly on legal text (10.27 vs. $\sim$6.6 for other methods), consistent with its lower QA accuracy observed in Table 1.

While AWQ demonstrates clear perplexity improvements over RTN on general-purpose benchmarks in the original work [3], the two methods show comparable perplexity in our domain-specific setting, suggesting that magnitude-based scaling provides limited additional benefit for specialized text.

We note that perplexity measures average next-token prediction quality across all tokens, whereas domain-specific QA accuracy depends on preserving the model's ability to reason over specialized vocabulary. Since domain-critical tokens constitute a sparse subset of the overall token distribution, perplexity is relatively insensitive to their degradation. This is why ASQ and AWQ exhibit similar perplexity despite ASQ's substantial advantage in downstream QA accuracy (Table 1), and motivates our choice of task accuracy as the primary evaluation metric.

Table 9: Held-out perplexity (↓) on domain-specific text (Llama-3-8B, W4A16). For each domain, 128 calibration samples are used for quantization and a disjoint set of 128 held-out samples from the same corpus is used for evaluation.

Method	**Finance**	**Legal**	**Medical**	**Avg.**
FP16 (Dense)	4.09	5.50	6.59	5.39
RTN	4.72	6.58	7.21	6.17
GPTQ	5.50	10.27	8.06	7.94
AWQ	4.75	6.59	7.25	6.20
ASQ (Ours)	4.78	6.62	7.20	6.20

F Related Work

F.1 Post-Training Quantization for LLMs

As Large Language Models (LLMs) continue to grow in scale, alleviating their computational overhead and inference latency has become essential for practical deployment [21–26]. Driven by the prohibitive retraining costs of large-scale models, Post-Training Quantization (PTQ) has become the primary approach for LLM compression. Existing PTQ settings are commonly categorized into W8A8 quantization [27, 2] and low-bit weight-only schemes such as W4A16 [1, 3].

To preserve accuracy under low-bit settings, a major direction in PTQ focuses on identifying and protecting salient weights. Mixed-precision approaches, such as SpQR [28], isolate outlier weights to retain in higher precision (e.g., FP16); however, their irregular memory access patterns often introduce hardware inefficiencies. In contrast, scaling-based methods like AWQ [3] apply channel-wise scaling before uniform INT4 quantization, maintaining hardware-friendly execution while effectively reducing quantization error.

Despite their architectural success, existing PTQ methods remain fundamentally "semantic-blind". They typically determine which weights or channels to protect using mechanical proxies, such as activation magnitudes or Hessian information, without considering the linguistic role of those components. As a result, channels that are numerically prominent but structurally driven may be over-protected, while channels that are activated only in semantically meaningful contexts may be under-protected. Recent studies such as AutoMixQ [29] and SliM-LLM [30] further highlight the importance of more carefully estimating quantization salience, but they still capture semantic importance only indirectly through statistical signals (e.g., hidden states or gradients) rather than explicit linguistic comprehension.

F.2 Functional Heterogeneity in LLM Representations

A growing body of work reveals that Transformer representations are not uniformly distributed across channels. Geva et al. [4] show that feed-forward layers act as key-value memories, with individual neurons encoding interpretable patterns. Dai et al. [5] further demonstrate that factual knowledge is localized in sparse "knowledge neurons" that activate selectively for specific relations.

Orthogonally, several studies identify a class of outlier channels that exhibit persistently large activations independent of input semantics. Kovaleva et al. [6] first characterize such dimensions in BERT, showing they dominate magnitude statistics despite carrying limited contextual information. Bondarenko et al. [7] trace this phenomenon to attention heads that learn "no-op" behavior, proposing architectural modifications to reduce outliers. He et al. [8] provide a theoretical account of how outlier features emerge during training through implicit regularization dynamics. Qiu et al. [9] unify attention sinks and residual outliers under a single framework, showing that such channels serve a rescaling function rather than encoding semantic content. From a quantization perspective, Dettmers et al. [27] demonstrate that a small number of outlier channels with extreme magnitudes can dominate quantization error, necessitating mixed-precision treatment. Sun et al. [31] observe a similar phenomenon in pruning, showing that combining weight magnitude with input activation norms yields better salience estimates than either signal alone.

These findings collectively establish that global activation magnitude conflates two fundamentally different roles: structural stabilization and semantic encoding. This distinction is precisely what existing quantization methods fail to exploit—they protect high-magnitude channels indiscriminately, regardless of whether those channels encode meaningful information. Our work leverages this insight by introducing a contrastive scoring mechanism that separates semantically selective channels from structurally dominant ones.

F.3 Agents in LLM Compression

The reasoning capabilities of recent LLMs have enabled agentic systems that integrate external tools, APIs, and environments to solve complex tasks. This intersection of agentic workflows and model compression has driven two primary research directions. The first compresses LLMs for tool-calling scenarios; for instance, TinyAgent [32] applies quantization to task-specific small language models deployed at the edge. The second leverages LLM agents to automate compression and deployment decisions, as exemplified by HAQA [33], which employs LLMs as black-box optimizers for hyperparameter search and bit-width selection.

While these approaches treat the agent either as the *target* of compression or as a *hyperparameter optimizer*, ASQ utilizes an LLM agent directly as a *Linguistic Expert* during the calibration phase. By employing the agent to semantically tag domain-critical tokens, our approach computes a contrastive semantic importance score. This establishes a novel agent-guided, semantic-aware PTQ framework that embeds linguistic understanding directly into the hardware-friendly scaling process.

G General-Text Calibration and Performance

While our main experiments focus on domain-specific calibration to preserve specialized knowledge, we further evaluate whether ASQ remains effective under general-text calibration. Specifically, we calibrate all methods on C4 (128 samples) and evaluate Llama-3-8B on five general benchmarks under W4A16 quantization.

As shown in Table 10, ASQ consistently achieves the best accuracy among all quantized methods across the five general benchmarks. Notably, AWQ and GPTQ also outperform RTN on every benchmark, recovering their expected advantage under this general-text calibration and evaluation setting.

These results demonstrate that ASQ's benefits are not limited to domain-specific calibration, as it consistently outperforms existing quantized baselines across all five general benchmarks under C4 calibration.

Table 10: General benchmark accuracy (%) under C4 general-text calibration (Llama-3-8B, W4A16). Bold indicates the best performance among quantized methods.

Method	**MMLU-G**	**ARC-E**	**ARC-C**	**PIQA**	**HellaSwag**
FP16 (Dense)	47.14	76.62	53.39	80.69	60.33
RTN	44.65	74.80	50.39	76.03	56.31
GPTQ	45.08	75.29	51.24	79.36	59.43
AWQ	45.21	76.06	51.47	79.14	59.42
ASQ (Ours)	**46.29**	**76.32**	**51.57**	**80.09**	**60.01**

Table 11: Pathway ablation on Llama-3-8B (W4A16). Accuracy (%) on five specialized-domain benchmarks. Bold indicates the best quantized method.

Method	FinanceQA	LexGLUE	MedQA	MedMCQA	PubMedQA	Avg.
FP16	46.63	25.58	42.81	36.94	62.80	42.95
AWQ (no semantic term)	40.57	26.63	36.53	35.52	61.80	40.21
Pathway A only	42.24	26.33	38.57	36.86	63.20	41.44
Pathway B only	42.64	26.61	39.14	36.58	63.40	41.67
A+B (ASQ, Ours)	**43.13**	**26.72**	**39.98**	**37.32**	**65.40**	**42.51**

H Additional Ablation Studies

We provide the full per-benchmark pathway and keyword-selector ablations under the Llama-3-8B W4A16 setting in Table 1, followed by sensitivity analyses of quantization group size and semantic boost strength. These results complement the summary in Section 6.6.

H.1 Component Ablation

Activation selectivity and attention routing. We evaluate Pathway A (contrastive activation selectivity), Pathway B (attention routing), and their combination on Llama-3-8B under the Table 1 setting. All ablated ASQ variants retain the AWQ magnitude-based scaling component. A-only adds activation-selectivity scoring; B-only adds attention routing at the output projections (W_O); and A+B combines both signals. The full per-benchmark results are shown in Table 11.

A-only and B-only improve average accuracy over AWQ by 1.23 pp and 1.46 pp, respectively. Combining the pathways improves further by 1.07 pp over A-only and 0.84 pp over B-only. These results show that activation selectivity and attention routing make individually beneficial and complementary contributions.

H.2 Keyword-Selection Ablation

To validate the semantic position set $\mathcal{K}$, we change only the keyword source while keeping the remaining ASQ pipeline identical, including the contrastive scoring and the α/β search grid. We compare agent-selected terms with coverage-matched random positions, a static domain dictionary, and spaCy named-entity recognition (NER). The random control matches the fraction of positions selected by the agent in each domain: 2.9% for finance, 1.1% for legal, and 10.4% for medical. This controls for positional coverage and tests whether the gain comes from selecting semantically meaningful positions rather than merely selecting a subset of positions.

Coverage-matched random positions do not improve over AWQ (40.06% vs. 40.21%), indicating that the additional semantic boost alone is insufficient. Static dictionaries and NER recover part of the benefit (41.71% and 41.76%, respectively), confirming the value of semantic position conditioning beyond the particular extractor. The agent-guided selector yields the highest average accuracy (42.51%), improving over AWQ by 2.30 pp and outperforming the dictionary and NER selectors by 0.80 pp and 0.75 pp, respectively, without requiring a manually curated domain lexicon or a domain-specific NER pipeline.

Table 12: Keyword-selector ablation on Llama-3-8B (W4A16). Accuracy (%) under an otherwise identical ASQ pipeline. Bold indicates the best quantized method.

Keyword source	FinanceQA	LexGLUE	MedQA	MedMCQA	PubMedQA	Avg.
FP16	46.63	25.58	42.81	36.94	62.80	42.95
AWQ (no semantic term)	40.57	26.63	36.53	35.52	61.80	40.21
Random positions (matched)	39.49	25.01	39.04	36.05	60.70	40.06
Static domain dictionary	42.59	25.64	**40.14**	36.39	63.80	41.71
spaCy NER	42.05	25.89	39.36	36.29	65.20	41.76
Agent (Ours)	**43.13**	**26.72**	39.98	**37.32**	**65.40**	**42.51**

Table 13: Group-size sensitivity of ASQ on Llama-3-8B (W4A16). Average accuracy (%) across five specialized benchmarks; FP16 accuracy is 42.95%.

Group size g	Avg. accuracy (%)
64	42.10
128	**42.51**
256	42.07

H.3 Hyperparameter Sensitivity

Quantization group size. We evaluate ASQ with group sizes $g \in \{64, 128, 256\}$, holding all other settings fixed (Llama-3-8B, Table 1 setting). The average accuracies across five specialized benchmarks are shown in Table 13.

Accuracy varies by at most 0.44 pp across the tested group sizes. Although $g = 128$ gives the highest result, both finer and coarser grouping retain comparable accuracy. We use $g = 128$, which is also a standard configuration in weight-only PTQ experiments.

Semantic boost strength β. We additionally perform a 17-point sweep over $\beta \in [0.03, 0.5]$, keeping the α search unchanged. Average accuracy varies by only 0.79 pp over the full sweep and by 0.38 pp within $\beta \leq 0.2$. The values selected by our calibration-based grid search ($\beta \approx 0.05$–0.07 in this setting) lie within this stable region. Thus, the observed gains are not attributable to an isolated favorable choice of the semantic boost strength.

I Calibration Overhead and Memory Analysis

All ASQ-specific computations take place during one-time offline calibration. Compared with AWQ, ASQ adds keyword extraction and position matching (or cached keyword loading and matching on subsequent runs), followed by one additional semantic-scoring sweep over the calibration set. Because the extracted keywords can be reused across runs, we separately report cached and uncached preprocessing costs. The measurements below follow the Table 1 calibration setting.

I.1 Incremental Calibration Time and GPU Memory

Incremental calibration time. Across the three domains (384 calibration samples in total), loading cached keywords and matching them to token positions required 108 s of CPU-side wall-clock time, or approximately 0.28 s per sample on average, and no external API calls. The subsequent semantic-score collection required one additional forward sweep over the calibration set, taking 23–25 s of GPU time across the three domains. For the initial uncached run, including external-API keyword extraction and position matching, the preprocessing step took 327 s in total across the three domains. The latter cost depends on the keyword extractor and can be reduced by a faster extractor or avoided on subsequent runs through caching. These figures describe the incremental ASQ-specific stages rather than the complete calibration-to-checkpoint wall-clock time of either method.

GPU memory and inference. Peak allocated GPU memory was 11.9 GiB per device for both AWQ and ASQ, with no measurable increase in this setting.

We measured peak allocation using PyTorch's `torch.cuda.max_memory_allocated()` function. After the learned scales are absorbed into the weights and adjacent operations, ASQ uses the same quantized linear operators as AWQ. Consequently, its quantized inference computation has the same structure as AWQ, with no additional operators, parameters, or runtime state. These statements describe the method's structure; the timings above measure incremental offline stages, not a separate end-to-end inference-latency test.

I.2 Hyperparameter Search Complexity

ASQ jointly searches nine α values and 21 β values over eight layer groups, resulting in $9 \times 21 \times 8 = 1{,}512$ candidate evaluations and 16 selected scalars per model–domain pair. In comparison, the official AWQ search structure evaluates approximately 128 input-sharing subsets over 20 ratios (about 2,560 evaluations), in addition to its clipping search. Thus, ASQ evaluates approximately 41% fewer grid candidates under these counting conventions, despite searching two hyperparameters. These counts characterize the search configurations and should not be interpreted as a corresponding reduction in end-to-end wall-clock time, since the candidate evaluations have different granularities and costs.

J Future Work and Limitations

ASQ opens a new direction by embedding linguistic understanding directly into the quantization process, and we view the present work as an initial step toward a broader class of *semantic-aware* compression methods. Several promising avenues lie ahead.

Beyond domain-level conditioning. While our framework conditions on domain-critical vocabulary, the same agent-guided principle naturally extends to finer-grained signals. Task-specific conditioning (e.g., reasoning chains, instruction-following patterns, or code semantics) and behavior-targeted compression (e.g., preserving safety-relevant or alignment-critical channels) are natural next steps. More broadly, semantic conditioning could be coupled with other compression paradigms beyond weight-only quantization, such as structured pruning, low-rank decomposition, or activation quantization, where channel-level salience is equally pivotal.

Current scope and limitations. Our experiments focus on the W4A16 weight-only setting with two model scales across three specialized domains, and broader quantization regimes remain to be explored. In addition, our self-quantization mode (Section 6.5) eliminates external agent dependency, but its effectiveness may diminish for very small target models whose linguistic capability is insufficient to reliably surface domain-critical terms; a systematic study across model scales would clarify the practical lower bound.

Toward broader applicability. We believe the agent-guided semantic-aware paradigm can evolve in several productive directions: integrating richer agent signals (e.g., multi-step reasoning traces or retrieval-augmented context), jointly optimizing scaling with mixed-precision allocation, and adapting the framework to multimodal models where "semantic salience" spans textual, visual, and structured inputs. We hope ASQ serves as a foundation for compression methods that are not only numerically faithful but also linguistically and behaviorally aware.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 explicitly state the four contributions (limitation identification, agent-guided conditioning, dual-pathway scoring, semantic-aware scaling), and the experimental results in Section 5 support these claims under the stated W4A16 setting.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix J, alongside future research directions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 4 provides a derivation of the proposed scaling rule under a diagonal approximation of the reconstruction error and standard uniform-quantization assumptions. All assumptions are stated inline alongside Eqs. 14–18, and each step is cross-referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 specifies models, baselines, calibration corpora, evaluation benchmarks, and quantization settings. Appendices A, B, C, and D provide preprocessing details, benchmark identifiers (with HuggingFace sources), the agent prompt template, and full hyperparameter search ranges.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets used are publicly available with citations (Section 5 and Appendix B). To preserve anonymity during review, code is not released; we plan to release the full implementation upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 and Appendix D report calibration sample size, sequence length, quantization configuration (W4A16, group size 128, symmetric), grid-search ranges for (α, β), layer grouping (G=8), random seed, and minimum keyword length.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following standard practice in PTQ literature (GPTQ, AWQ, SmoothQuant), we report single-run results with a fixed random seed (42) for calibration sampling. PTQ is deterministic given fixed calibration data, and the dominant variance source is calibration sample selection, which we control via fixed seed.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are reported in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets and pre-trained models, involves no human subjects, and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Positive impacts include reduced memory and energy costs of LLM deployment, lowering the barrier for on-device and resource-constrained inference (e.g., specialized domains such as healthcare). The method is a compression technique applied to existing pretrained models and does not introduce new generation capabilities; potential negative impacts are inherited from the underlying base models rather than introduced by ASQ.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release new pretrained models or scraped datasets. The proposed method is a quantization technique applied to existing publicly released models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All models (Llama-3-8B, Qwen2.5-32B, Gemini-2.5-Flash) and datasets (PMC-Patients V2, Pile of Law, EDGAR-CORPUS, MedQA, MedMCQA, PubMedQA, LexGLUE, MMLU) are properly cited (Section 5, Appendix A, Appendix B) and used in accordance with their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. Code release is planned upon acceptance, with documentation included.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: An LLM (Gemini-2.5-Flash) is used as a core methodological component for semantic keyword extraction, as described in Section 3 and Appendix C. We additionally analyze a self-quantization variant (Section 6.5) where the target model itself serves as the agent.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.