Coverage-Based Calibration for Post-Training Quantization via Weighted Maximum Coverage over Outlier Channels

Sanjeda Akter[1*] **Ibne Farabi Shihab**[1*] **Anuj Sharma**[2]

[1]Department of Computer Science
[2]Department of Civil, Construction and Environmental Engineering
Iowa State University

Abstract

Post-training quantization (PTQ) compresses large language models using a small calibration set, making sample selection consequential for low-bit deployment. We study calibration selection for weight-only PTQ, where data determines activation-conditioned weight reconstruction and scale search. We formulate selection as weighted maximum coverage over module-level input channels with rare, high-magnitude activations. Each sequence covers the channels it activates above a reference threshold, with weights combining activation magnitude and squared weight-column norm. The objective is monotone submodular, so greedy selection admits the classical $(1-1/e)$ approximation guarantee for coverage. We instantiate this formulation in CoverCal, a backend-agnostic selector operating on cached activation masks. On LLaMA-3-8B at $K = 128$, CoverCal improves MMLU by 1.4 points over random calibration under AWQ and 1.6 under GPTQ, and by 0.5 points over ZipCal under both backends. Controlled interventions connect weighted event selection to AWQ scale-search outcomes and GPTQ activation statistics. Additional experiments cover instruction-tuned Llama, Mistral, and Qwen models, a 70B checkpoint, and a second candidate corpus. The gains are empirical and most useful at small calibration budgets: larger random sets approach the same quality, while CoverCal reaches that regime with less measured calibration time, including profiling.

1 Introduction

Post-training quantization (PTQ) is a leading strategy for deploying large language models (LLMs) under memory and latency constraints. Modern weight-only PTQ methods such as GPTQ [8] and AWQ [14] quantize from FP16 to INT4 using a small calibration set and no retraining. The calibration set determines the activation-conditioned statistics used by the quantizer: GPTQ uses calibration activations through a layerwise reconstruction/Hessian approximation, while AWQ uses activation magnitudes to identify salient weight channels and search protective scaling factors. Thus calibration selection directly affects the quantized weights even though the deployed W4A16 model does not quantize or clip activations at inference.

In standard practice, calibration samples are usually chosen by count, often uniformly from web-text corpora. This practice is fragile because calibration is not only a problem of approximating the average input distribution. Transformer activations contain rare but high-magnitude coordinates [5, 28, 14]. When a calibration set misses such coordinates for a quantized linear module, the backend observes an incomplete activation-conditioned reconstruction problem: the diagonal energy, saliency, and

*Equal contribution.

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

scale-search evidence associated with the missed coordinate are too small relative to a reference pool. The resulting weight quantization error can concentrate on a small number of channels and modules even when the calibration text looks distributionally plausible.

We therefore treat calibration selection as a coverage problem. The objects to cover are not only documents or topics, but module-level outlier channels: triples (l, m, c) consisting of layer l, quantized linear module m in that layer, and input channel c of that module. This definition is important for LLMs. A transformer block contains several quantized matrices (attention projections, output projections, and MLP projections) with different input dimensions and different activation distributions. A single residual-stream channel set is not sufficient to describe the columns quantized by AWQ or GPTQ. CoverCal records, for each candidate sequence, which module-level channels exceed a reference outlier threshold at any non-padding token position. It then greedily selects sequences that maximize newly covered weighted mass.

The relevant events need not be common to matter. At $K = 128$ on LLaMA-3-8B, CoverCal covers 27.4% of module events but 84.6% of their total weight. We test whether this concentration is useful through selector-level interventions and measurements of the resulting AWQ scales and GPTQ activation statistics (Section 6). These comparisons use the same candidate pool and calibration budget.

Once calibration is stated as weighted coverage, the algorithmic problem is standard. Each candidate sample covers the outlier channels it activates, each channel carries a nonnegative weight reflecting a bound on its possible reconstruction contribution, and the task is to select at most K samples maximizing total covered weight. The objective is monotone submodular, so greedy selection inherits the $(1 - 1/e)$ approximation guarantee of [20]. The guarantee is for the coverage objective, not for downstream accuracy. The empirical question is whether this objective is a better calibration proxy than generic representativeness or single-sample hardness.

Our contributions are threefold: (i) a calibration objective over module/input-channel triples, including post-SwiGLU down-projection inputs (Section 3); (ii) a greedy selector with the standard approximation guarantee for nonnegative weighted coverage (Theorem 3.5 and Algorithm 1); and (iii) evidence that this objective is a useful calibration signal in the tested AWQ/GPTQ settings (Sections 5 and 6). CoverCal is not a new quantizer or a model-agnostic selector: the selected texts are supplied to unmodified AWQ and GPTQ implementations, and the selector uses checkpoint-specific profiling. Its theoretical guarantee concerns coverage, while reconstruction and downstream quality are evaluated empirically.

2 Related work

Weight-only LLM quantization. GPTQ [8] quantizes weights layer by layer using approximate second-order information from calibration activations, while AWQ [14] uses activation magnitudes to identify salient channels and protect them during quantization. SqueezeLLM [13], QuIP [3], and SpQR [6] modify the representation or error allocation of low-bit weights. Rotation-based weight-quantization methods such as QuaRot [1] and SpinQuant [15] use fixed or learned rotations. TurboQuant [31] studies related rotation-based vector quantization, including KV-cache compression. These approaches change the quantizer or representation; CoverCal instead selects calibration samples for a fixed weight-only backend.

Activation outliers. Activation outliers in transformers were documented in LLM.int8() [5]. SmoothQuant [28] migrates activation difficulty into weights through channel rescaling, and AWQ [14] uses activation magnitude as a weight-saliency signal. Work on quantizable transformers and outlier formation studies when and why such coordinates arise [2]. The present work does not remove outliers. It asks which calibration samples should be used when the backend remains sensitive to rare high-magnitude activation coordinates.

Calibration-data selection. Studies of calibration source and count [26], SelectQ's clustering-based selection [32], self-calibration from generated data [27], ZipCal's model-agnostic lexical curation [19], and capability-preserving calibration [9] show that calibration data matters. SelectQ and ZipCal both select from a fixed pool and are included in our direct comparison. SelectQ targets activation-distribution representativeness, ZipCal targets lexical-frequency diversity, and

COVERCAL targets weighted module-event coverage. Methods that generate data or change the candidate distribution require a different comparison protocol.

Submodular selection. Weighted set cover and greedy approximation are classical [4, 20]. Submodular objectives also appear in active learning [23], coreset construction [22, 18], and data selection for neural-network training [29, 12]. The novelty here is not the greedy algorithm. It is the identification of the objects and weights to cover for LLM PTQ: module-level activation outlier events whose absence distorts activation-conditioned quantization statistics.

3 Problem formulation

This section defines calibration selection at the level actually used by weight-only transformer quantizers. We first write the module-level reconstruction objective, then define outlier coverage over module/input-channel triples, and finally prove a surrogate consistency result that uses squared weight-column norms, matching the squared Frobenius objective.

3.1 Module-level PTQ reconstruction

Let f_θ be a transformer with layers $l \in \{1, \dots, L\}$. Let $\mathcal{M}_l$ denote the quantized linear modules in layer l; for LLaMA/Mistral-style decoder blocks this includes attention projections q, k, v, o and MLP projections $\mathrm{gate}, \mathrm{up}, \mathrm{down}$. For a module $m \in \mathcal{M}_l$, write

$$\mathbf{W}^{(l,m)} \in \mathbb{R}^{d_{\mathrm{out}}^{(l,m)} \times d_{\mathrm{in}}^{(l,m)}} \tag{1}$$

for its FP16 weight matrix and

$$\mathbf{A}^{(l,m)}(\mathcal{S}) \in \mathbb{R}^{d_{\mathrm{in}}^{(l,m)} \times T(\mathcal{S})} \tag{2}$$

for the matrix of input activations to that module, concatenated over non-padding token positions in calibration set $\mathcal{S}$. The module-level reconstruction loss optimized or approximated by common weight-only PTQ backends has the form

$$\mathcal{R}_{l,m}(\hat{\mathbf{W}}; \mathcal{S}) = \left\| \mathbf{W}^{(l,m)} \mathbf{A}^{(l,m)}(\mathcal{S}) - \hat{\mathbf{W}} \mathbf{A}^{(l,m)}(\mathcal{S}) \right\|_F^2 . \tag{3}$$

Equivalently, with $\Delta^{(l,m)} = \mathbf{W}^{(l,m)} - \hat{\mathbf{W}}^{(l,m)}$ and empirical activation covariance

$$\mathbf{H}_{\mathcal{S}}^{(l,m)} = \frac{1}{T(\mathcal{S})} \mathbf{A}^{(l,m)}(\mathcal{S}) \mathbf{A}^{(l,m)}(\mathcal{S})^\top , \tag{4}$$

we have

$$\frac{1}{T(\mathcal{S})} \mathcal{R}_{l,m}(\hat{\mathbf{W}}; \mathcal{S}) = \mathrm{Tr}\left(\Delta^{(l,m)} \mathbf{H}_{\mathcal{S}}^{(l,m)} \Delta^{(l,m)\top} \right) . \tag{5}$$

This expression makes explicit how calibration samples enter weight-only PTQ: they determine the activation covariance, saliency, and scale-search evidence against which weight quantization error is evaluated. No inference-time activation clipping assumption is needed.

3.2 Token-level outlier channels

For a candidate sequence s and module/channel triple $q = (l, m, c)$, define the token-max activation magnitude

$$a_q(s) = \max_{t \in \mathcal{T}(s)} \left| A_{c,t}^{(l,m)}(s) \right| , \tag{6}$$

where $\mathcal{T}(s)$ excludes padding tokens. For attention q, k, v projections, $A^{(l,m)}$ is the hidden state after the attention input RMSNorm and before the projection. For the output projection o, it is the attention output before o_proj. For MLP gate/up projections, it is the hidden state after the MLP input RMSNorm. For the down projection, it is the SwiGLU product supplied to $\mathrm{down_proj}$. RoPE is applied after q/k projection and therefore does not change the input channels of q_proj or k_proj; GQA/MQA changes the output dimensions of K/V projections but not the input-channel coverage definition. KV-cache behavior is unaffected because COVERCAL changes only calibration selection for weight quantization.

Let $\mathcal{P}$ be the candidate pool and also the reference pool unless stated otherwise. For each module m in each layer l, compute a threshold from token-max magnitudes

$$\tau^{(l,m)} = \mu^{(l,m)} + 6\sigma^{(l,m)}, \qquad \mu^{(l,m)}, \sigma^{(l,m)} \text{ computed over all } a_{(l,m,c)}(s), \tag{7}$$

where $s \in \mathcal{P}$ and $1 \le c \le d_{\text{in}}^{(l,m)}$. We use 6σ throughout the main comparisons. Section E reports a 5σ–7σ sensitivity study; this tests local robustness rather than a universal threshold.

Definition 3.1 (Module-level outlier universe)**.** The outlier universe is

$$\mathcal{C} = \left\{ (l,m,c) : \max_{s \in \mathcal{P}} a_{(l,m,c)}(s) > \tau^{(l,m)} \right\}. \tag{8}$$

A calibration set $\mathcal{S} \subseteq \mathcal{P}$ covers $q = (l,m,c) \in \mathcal{C}$ if

$$\max_{s \in \mathcal{S}} a_q(s) > \tau^{(l,m)}. \tag{9}$$

The covered set is

$$\text{Cov}(\mathcal{S}) = \{ q \in \mathcal{C} : \max_{s \in \mathcal{S}} a_q(s) > \tau^{(l,m)} \}. \tag{10}$$

Because $\mathcal{P}$ is the reference pool, every $q \in \mathcal{C}$ is coverable by at least one candidate sequence. The remaining gap between $\text{Cov}(\mathcal{S})$ and $\mathcal{C}$ is therefore due to the calibration budget K, not to missing candidates.

3.3 The weighted coverage objective

Coverage alone treats all outliers equally. To connect coverage to squared reconstruction error, assign each outlier triple $q = (l,m,c)$ the weight

$$w_q = \gamma_m \left(\frac{o_q^{\text{ref}}}{\tau^{(l,m)}} \right)^2 \left\| \mathbf{W}_{:,c}^{(l,m)} \right\|_2^2, \qquad o_q^{\text{ref}} = \max_{s \in \mathcal{P}} a_q(s), \tag{11}$$

where $\gamma_m \ge 0$ is an optional backend/module coefficient set to 1 in the main experiments. The squared normalized magnitude reflects the largest observed tail event, and the squared column norm is dimensionally consistent with the squared output norm: for a scalar channel perturbation u_c, the output contribution through column c is $\|\mathbf{W}_{:,c}^{(l,m)}\|_2^2 u_c^2$.

The calibration selection problem is

$$\mathcal{S}^* = \arg \max_{\mathcal{S} \subseteq \mathcal{P},\ |\mathcal{S}| \le K} F(\mathcal{S}), \qquad F(\mathcal{S}) = \sum_{q \in \text{Cov}(\mathcal{S})} w_q. \tag{12}$$

Proposition 3.2 (Monotonicity and submodularity)**.** *If all weights w_q are nonnegative, then F is monotone submodular.*

Proof. For any $\mathcal{S} \subseteq \mathcal{P}$ and $s \in \mathcal{P}$, adding s can only add covered triples, never remove them, so $F(\mathcal{S} \cup \{s\}) \ge F(\mathcal{S})$. For submodularity, let $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{P}$ and $s \in \mathcal{P} \setminus \mathcal{B}$. The marginal gain of s under $\mathcal{A}$ is the weight of triples covered by s but not by $\mathcal{A}$; the marginal gain under $\mathcal{B}$ is the weight of triples covered by s but not by $\mathcal{B}$. Since $\text{Cov}(\mathcal{A}) \subseteq \text{Cov}(\mathcal{B})$, the former set contains the latter. Nonnegativity of w_q gives diminishing returns. □

3.4 A weight-only surrogate bound

The objective in Equation (12) is not claimed to upper-bound downstream accuracy loss. It is a tractable proxy for a specific calibration failure: omitted tail activation events produce understated activation-conditioned sensitivity for the corresponding weight columns. We formalize this proxy in a way that matches weight-only PTQ.

For a module/channel triple $q = (l,m,c)$, define a residual miss factor $r_q(\mathcal{S}) \in [0,1]$ measuring how much reference-pool tail sensitivity remains unrepresented by $\mathcal{S}$. We assume

$$q \in \text{Cov}(\mathcal{S}) \Rightarrow r_q(\mathcal{S}) \le \eta_q, \qquad 0 \le \eta_q \le 1, \tag{13}$$

$$q \notin \text{Cov}(\mathcal{S}) \Rightarrow r_q(\mathcal{S}) \le 1. \tag{14}$$

The idealized zero-residual model is the special case $\eta_q = 0$. In practice η_q is not zero because one observed tail token does not remove rounding error or guarantee that larger future activations cannot occur. Keeping η_q explicit avoids the false claim that single-event coverage eliminates all error.

Define the missed-tail surrogate

$$\mathcal{L}_{\text{tail}}(\mathcal{S}) = \sum_{q \in \mathcal{C}} w_q r_q(\mathcal{S}). \tag{15}$$

This quantity can be interpreted as a diagonal upper envelope for the portion of Equation (5) attributable to activation-tail events that are present in the reference pool but weakly represented in the calibration set. The squared column norm in w_q is the correct scale for squared output error; the normalized activation factor is a finite-pool proxy for tail energy.

Proposition 3.3 (Coverage controls the missed-tail surrogate)**.** *Under Equations (13) and (14),*

$$\mathcal{L}_{\text{tail}}(\mathcal{S}) \le \sum_{q \in \mathcal{C} \setminus \text{Cov}(\mathcal{S})} w_q + \sum_{q \in \text{Cov}(\mathcal{S})} \eta_q w_q. \tag{16}$$

In the idealized case $\eta_q = 0$ for all q, maximizing $F(\mathcal{S})$ is equivalent to minimizing the surrogate scale $\sum_{q \in \mathcal{C}} w_q - F(\mathcal{S})$.

Proof. Split the sum in Equation (15) into covered and uncovered triples. If q is covered, Equation (13) gives $w_q r_q(\mathcal{S}) \le \eta_q w_q$. If q is uncovered, Equation (14) gives $w_q r_q(\mathcal{S}) \le w_q$. Summing both parts gives Equation (16). Setting all $\eta_q = 0$ yields $\mathcal{L}_{\text{tail}}(\mathcal{S}) \le \sum_{q \notin \text{Cov}(\mathcal{S})} w_q = \sum_q w_q - F(\mathcal{S})$. □

Remark 3.4 (Connection to GPTQ and AWQ). For GPTQ, Equation (5) shows that the diagonal term for channel c scales as $\|\Delta_{:,c}^{(l,m)}\|_2^2 H_{\mathcal{S},cc}^{(l,m)}$ plus cross-channel terms. Missing high-magnitude tokens biases H_{cc} and associated blockwise updates downward relative to the reference pool. For AWQ, activation magnitudes determine which channels are salient and how protective scales are searched; missing high-magnitude module inputs weakens that saliency signal. The bound above is therefore not a backend proof, but it is aligned with the quantities each backend actually uses. It deliberately avoids the incorrect interpretation that W4A16 AWQ/GPTQ calibration defines an inference-time activation clipping scale.

Theorem 3.5 (Greedy approximation)**.** *Let $\mathcal{S}_K^*$ be an optimal solution of Equation (12), and let $\mathcal{S}_K^{\text{greedy}}$ be the greedy set that repeatedly adds the candidate with largest marginal weighted coverage. Then*

$$F(\mathcal{S}_K^{\text{greedy}}) \ge (1 - 1/e) F(\mathcal{S}_K^*). \tag{17}$$

Proof. This is the standard cardinality-constrained monotone-submodular maximization guarantee of [20], applied to the nonnegative coverage objective above.

□

4 Method

CoverCal consists of a one-time profiling phase and a CPU-side greedy selection phase. Profiling records module-input activations from a fixed candidate pool; selection operates only on the resulting sparse binary coverage matrix and weights.

Activation hooks. For each model, we attach hooks to the input of every quantized linear module. For LLaMA/Mistral-style blocks these are: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj. The activation for down_proj is measured after the SwiGLU product, not on the residual stream. Shared-input projections have identical binary masks: $M_q = M_k = M_v$ and $M_{\text{gate}} = M_{\text{up}}$. Their weights can differ because their weight-column norms differ. Table 1 gives the resulting module-resolved event counts on LLaMA-3-8B: of 1,245,184 possible module-input coordinates, 47,025 are events at 6σ. A layer-level key over the $32 \times 4096 = 131{,}072$ residual-stream coordinates gives only 4,950 events and cannot represent the 458,752 post-SwiGLU down-projection inputs or the module-specific weights in Equation (11). All coverage, threshold, and selector-dependent results use the module-qualified key (Section G).

Table 1: LLaMA-3-8B module-event decomposition at 6σ. Raw coordinates and events precede duplication across modules sharing an input tensor.

Input / modules	Raw coordinates	Raw events	Copies	Expanded coordinates	Expanded events
Attention RMSNorm / q, k, v	131,072	5,912	3	393,216	17,736
Attention output / o	131,072	5,864	1	131,072	5,864
MLP RMSNorm / gate, up	131,072	5,701	2	262,144	11,402
Post-SwiGLU / down	458,752	12,023	1	458,752	12,023
Total	–	–	–	1,245,184	47,025

Algorithm 1 CoverCal: greedy weighted module-coverage selection.

Require: Candidate pool $\mathcal{P} = \{s_i\}_{i=1}^{N}$, binary coverage matrix $M \in \{0,1\}^{N \times |\mathcal{C}|}$, weights $w \in \mathbb{R}_+^{|\mathcal{C}|}$, budget K, tie-breaking seed r.
Ensure: Calibration set $\mathcal{S}$ with $|\mathcal{S}| = K$.

$\mathcal{S} \leftarrow \emptyset$; $\texttt{selected} \leftarrow \mathbf{0} \in \{0,1\}^N$
$\texttt{covered} \leftarrow \mathbf{0} \in \{0,1\}^{|\mathcal{C}|}$
for $k = 1, \ldots, K$ **do**
  for each index i with $\texttt{selected}_i = 0$ **do**
    $\texttt{gain}_i \leftarrow \sum_{j:\, \texttt{covered}_j = 0} M_{i,j} w_j$
  end for
  $i^* \leftarrow \arg\max_{i:\, \texttt{selected}_i = 0} \texttt{gain}_i$ with deterministic seed-r tie breaking
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_{i^*}\}$; $\texttt{selected}_{i^*} \leftarrow 1$
  $\texttt{covered}_j \leftarrow 1$ for all j such that $M_{i^*,j} = 1$
end for
return $\mathcal{S}$

Coverage matrix. Given $\mathcal{P} = \{s_1, \ldots, s_N\}$, we compute $a_q(s_i)$ from Equation (6), thresholds from Equation (7), and universe $\mathcal{C}$. We then assemble

$$M \in \{0,1\}^{N \times |\mathcal{C}|}, \qquad M_{i,j} = \mathbf{1}[a_{q_j}(s_i) > \tau^{(l_j, m_j)}], \tag{18}$$

where $q_j = (l_j, m_j, c_j)$ is the jth outlier triple. We store M as a sparse CSR matrix and store the weight vector $w \in \mathbb{R}_+^{|\mathcal{C}|}$. The weighted row for sample i is never materialized densely; marginal gains are computed as sparse dot products against an uncovered mask.

Selection algorithm. The naive complexity is $O(KN|\mathcal{C}|)$; sparse rows and an uncovered bitmask avoid dense evaluation. On LLaMA-3-8B with $N = 10{,}000$ and $K = 128$, profiling takes 15.1 minutes on one A100 and CPU selection takes 8.4 seconds. The profile can be reused across budgets and backends only when the checkpoint, tokenizer, candidate pool, maximum length, and hook definition are unchanged. Section F includes profiling in the one-shot cost.

5 Experiments

We evaluate whether weighted module-event coverage improves calibration selection when the quantizer and candidate pool are fixed. The main comparison uses LLaMA-3-8B; additional experiments test model family, parameter scale, instruction tuning, and candidate corpus.

Setup and baselines. The main pool contains $N = 10{,}000$ RedPajama sequences [25], with maximum length 2048. Additional tests cover Llama-2/3 [24, 17] and Mistral [11]. We use AWQ and GPTQ INT4 with group size 128. Random, Max-PPL, Max-ActVar, Stratified, SelectQ, ZipCal, and CoverCal use identical candidate IDs, budget, tokenizer, backend settings, and evaluation harness within each comparison. SelectQ treats each sequence as one sample, averages activation statistics over non-padding tokens, keeps its layer-wise dynamic clustering and Knowledge Distance, and selects the nearest real sequence to each selected centroid (Section A). ZipCal uses its lexical-frequency diversity criterion on the same candidates.

Table 2: Fixed-pool comparison on LLaMA-3-8B, $K = 128$. Accuracy is in percent; perplexity is on WikiText-2. Downstream values are mean $\pm$ standard deviation over five paired calibration seeds. Raw and weighted coverage use the module-event universe $\mathcal{C}$.

Selector	Raw coverage (%)	Weighted coverage (%)	AWQ MMLU↑	GPTQ MMLU↑	GPTQ PPL↓
Random	17.1	42.8	63.1 ± 0.3	62.5 ± 0.3	6.53 ± 0.06
Max-PPL	10.2	21.6	63.4 ± 0.2	62.8 ± 0.2	6.50 ± 0.05
Max-ActVar	10.8	25.1	63.8 ± 0.2	63.2 ± 0.2	6.46 ± 0.04
Stratified	16.4	38.9	63.5 ± 0.2	63.0 ± 0.2	6.48 ± 0.05
SelectQ	21.3	56.2	63.9 ± 0.2	63.4 ± 0.2	6.42 ± 0.05
ZipCal	22.7	59.4	64.0 ± 0.2	63.6 ± 0.2	6.38 ± 0.04
CoverCal	**27.4**	**84.6**	$\mathbf{64.5 \pm 0.2}$	$\mathbf{64.1 \pm 0.2}$	$\mathbf{6.28 \pm 0.03}$

Table 3: Instruction-tuned models at $K = 128$. ZipCal is the strongest non-CoverCal selector in each displayed setting. Entries are reported mean $\pm$ standard deviation; IFEval uses strict prompt-level accuracy. FP16 reference scores are in Section C.

	MMLU↑		IFEval↑	
Model / backend	ZipCal	CoverCal	ZipCal	CoverCal
Llama-3-8B-Instruct / AWQ	62.4 ± 0.2	63.1 ± 0.2	76.0 ± 0.4	76.8 ± 0.3
Mistral-7B-Instruct / GPTQ	59.1 ± 0.2	59.7 ± 0.2	54.0 ± 0.5	54.7 ± 0.4
Qwen2.5-7B-Instruct / AWQ	70.2 ± 0.2	70.8 ± 0.2	81.5 ± 0.4	82.1 ± 0.3
Llama-3-70B-Instruct / AWQ	78.3 ± 0.1	78.8 ± 0.1	84.2 ± 0.2	84.7 ± 0.2

Evaluation and uncertainty. We report MMLU 5-shot accuracy [10], WikiText-2 perplexity [16], and strict prompt-level IFEval accuracy [33] for instruction-tuned models. The main selector comparison and mechanism controls use five paired calibration seeds. Errors denote standard deviation where reported. Paired evaluation-unit bootstrap intervals in Section B quantify evaluation-set uncertainty conditional on those five seed pairs. Exact reproduction artifacts are listed in Section J.

5.1 Fixed-pool comparison

Table 2 compares all seven selectors at $K = 128$. CoverCal reaches 64.5 MMLU under AWQ and 64.1 under GPTQ, exceeding Random by 1.4 and 1.6 points and the strongest displayed comparator, ZipCal, by 0.5 points under each backend. GPTQ perplexity improves by 0.10 relative to ZipCal. Paired evaluation-unit bootstrap intervals for these differences are $[+0.34, +0.66]$ and $[+0.40, +0.60]$ MMLU points and $[-0.12, -0.08]$ perplexity (Section B). These results establish the comparison at this operating point; the coverage guarantee does not imply a universal ordering of selectors.

5.2 Model, scale, and corpus coverage

Table 3 extends the comparison to instruction-tuned Llama, Mistral, and Qwen2.5 [30], including a 70B checkpoint. The instruction pool contains 10,000 UltraChat-200k training examples [7], rendered with each checkpoint's official chat template. All selectors use identical candidate IDs within each comparison. CoverCal improves both MMLU and IFEval over ZipCal in these four settings. A matched C4 pool [21] also gives a gain on LLaMA-3-8B: 64.3 versus 63.9 MMLU under AWQ. Section C separates corpus choice, instruction-pool alignment, and transfer of selected texts. On Llama-3-8B-Instruct, CoverCal reaches 62.7 MMLU when selecting from RedPajama and 63.1 from the instruction pool, where Random reaches 62.4: an aligned pool helps, and CoverCal retains a gain within it. Texts selected on a base checkpoint transfer partially to its instruction-tuned target, but target-specific profiling is best in all four settings.

5.3 Budget, runtime, and threshold sensitivity

At $K = 64$, CoverCal reaches 64.0 MMLU on LLaMA-3-8B under AWQ, exceeding Random at $K = 256$ (63.8). Larger random sets approach the same quality: Random at $K = 1024$ reaches 64.4 ± 0.1, compared with 64.5 ± 0.2 for CoverCal at $K = 128$. Including profiling, selection,

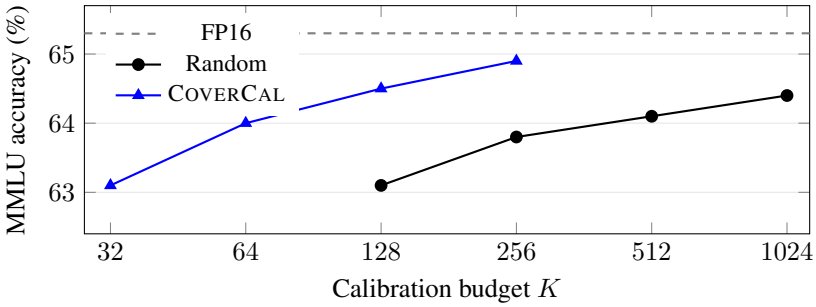

Figure 1: Budget curve on LLaMA-3-8B under AWQ. Points show reported means; variability for the timed operating points is given in Table 4. Larger random sets approach the quality of smaller CoverCal sets.

Table 4: One-shot AWQ calibration cost and MMLU on LLaMA-3-8B. Total time includes profiling, selection, and quantization, but excludes downstream evaluation.

Selector / budget	Profiling	Selection	Quantization	Total	MMLU
Random / 128	0	0	15.2 min	15.2 min	63.1 ± 0.3
Random / 256	0	0	28.4 min	28.4 min	63.8 ± 0.2
Random / 512	0	0	52.6 min	52.6 min	64.1 ± 0.2
Random / 1024	0	0	103.4 min	103.4 min	64.4 ± 0.1
CoverCal / 64	15.1 min	8.4 s	7.9 min	23.1 min	64.0 ± 0.2
CoverCal / 128	15.1 min	8.4 s	15.2 min	30.4 min	64.5 ± 0.2

and quantization, these runs take 103.4 and 30.4 minutes, respectively (Table 4). This is a measured calibration-budget and runtime tradeoff; it does not establish statistical equivalence or an inference-time speedup. Figure 1 shows the budget curve. Charging each selector for its required features, selection overhead is 31.6 seconds for ZipCal, 21.8 minutes for SelectQ, and 15.24 minutes for CoverCal (Table 14).

The 5σ–7σ sweep changes MMLU by at most 0.1 points within each tested 7B/8B or 13B model (Table 5; event counts and selected-set overlap are in Section E). The 70B threshold audit measures event counts and selected-set overlap only. We therefore claim local stability in the measured settings, rather than threshold invariance at every scale.

6 Analysis

Which events are covered? CoverCal covers 27.4% of module events but 84.6% of their weight. The top 1%, 5%, and 10% of events carry 35.2%, 62.7%, and 76.9% of total weight. This concentration explains why high raw coverage is unnecessary for high weighted coverage. Across the seven selectors, coverage and AWQ MMLU have Spearman correlation $\rho = 0.75$, but their rankings are not identical. With only seven methods, this is descriptive evidence for an imperfect proxy.

Can rare tokens change backend statistics? A maximum above 6σ need not materially affect an average over many tokens: at $K = 128$, AWQ can average over up to $128 \times 2048 = 262{,}144$ token activations per channel. We therefore measure the largest token's share of the channel's absolute-activation sum and squared-activation sum (Section H). Among top-weight-decile events, their 90th-percentile shares are 5.3% and 19.6%, respectively. The effect is concentrated in the upper tail; the median event contributes much less.

Do event identity and weight matter? We perturb the selector before reselection while fixing the candidate pool, K, backend, and paired seed. Shuffling weights preserves the weight multiset but changes event identity. High-weight removal removes columns uniquely supplied by the unperturbed solution; a matched low-weight removal provides a negative control. Table 6 shows that shuffling and high-weight removal worsen both backend statistics and downstream scores. Unweighted coverage

Table 5: MMLU (%) in the threshold sweep under AWQ INT4 at $K = 128$. Errors denote reported standard deviations.

Model	5σ	6σ	7σ
Llama-3-8B	64.4 ± 0.2	64.5 ± 0.2	64.4 ± 0.2
Mistral-7B	61.9 ± 0.2	62.0 ± 0.2	61.9 ± 0.2
Qwen2.5-7B	70.7 ± 0.2	70.8 ± 0.2	70.7 ± 0.2
Llama-2-13B	54.5 ± 0.2	54.6 ± 0.2	54.5 ± 0.2

Table 6: Selector-level controls on LLaMA-3-8B at $K = 128$, mean $\pm$ standard deviation over five paired seeds. Scale and diagonal errors are relative to the full-pool reference (Section H); smaller is better.

Selection condition	AWQ scale error↓	INT4-code disagr. (%)↓	GPTQ diagonal error↓	AWQ MMLU↑	GPTQ MMLU↑
Random	0.168 ± 0.009	9.8 ± 0.5	0.224 ± 0.008	63.1 ± 0.3	62.5 ± 0.3
Length/domain diversity	0.139 ± 0.007	8.2 ± 0.4	0.187 ± 0.007	63.6 ± 0.2	63.0 ± 0.2
Unweighted coverage	0.111 ± 0.006	6.9 ± 0.3	0.153 ± 0.006	64.0 ± 0.2	63.6 ± 0.2
Weights shuffled	0.146 ± 0.008	8.7 ± 0.4	0.194 ± 0.007	63.5 ± 0.2	63.0 ± 0.2
High-weight removal	0.132 ± 0.007	7.9 ± 0.4	0.181 ± 0.006	63.7 ± 0.2	63.3 ± 0.2
Low-weight removal	0.074 ± 0.005	4.8 ± 0.3	0.121 ± 0.005	64.4 ± 0.2	64.0 ± 0.2
CoverCal	0.071 ± 0.004	4.6 ± 0.2	0.118 ± 0.004	64.5 ± 0.2	64.1 ± 0.2

reaches 64.0 AWQ MMLU, between length/domain-matched diversity (63.6) and weighted coverage (64.5). Section H adds AWQ salience agreement and GPTQ perplexity for every control.

CoverCal also matches the full-pool AWQ scale-search grid choice in $84.2\% \pm 1.3\%$ of modules, versus $58.4\% \pm 2.1\%$ for Random, and reduces INT4-code disagreement from 9.8% to 4.6%. Matched low-weight removal leaves MMLU close to the unperturbed result. Together, these controls connect weighted event selection to actual backend behavior beyond generic diversity. They do not imply that one token eliminates a channel's quantization error.

7 Discussion and conclusion

CoverCal formulates calibration selection as weighted coverage over module-level outlier events. Its guarantee applies to that objective; improvements in AWQ/GPTQ quality are empirical. The direct selector comparisons, backend measurements, and broader evaluations support weighted event coverage as a useful signal in the tested dense decoder-only models.

Limitations. Profiling is checkpoint-specific and depends on the candidate pool: unseen events cannot be covered, and poor pool alignment can limit the benefit. Selected texts transfer partially between corresponding base and instruction-tuned checkpoints, but target-specific profiling performs best (Section C). Our experiments do not establish generalization to mixture-of-experts, encoder-decoder, or multimodal models. The surrogate omits cross-channel covariance and backend optimization details, and larger random sets can approach the same quality at practical budgets. The measured runtime benefit concerns the upfront calibration procedure and may matter less when that cost is small relative to deployment.

Broader impact. Better calibration can preserve quality at lower deployment cost, but aggregate benchmark scores do not establish uniform preservation across domains or groups. Quantized models retain the capabilities and potential harms of their source models. Deployment should include application-relevant domain and subgroup evaluation.

Acknowledgments and Disclosure of Funding

Funding. The authors received no external funding specifically supporting this work.

Competing Interests. The authors declare no competing interests.

References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[2] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.

[3] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36:4396–4429, 2023.

[4] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[5] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

[6] 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, 2023. URL https://arxiv.org/abs/2306.03078.

[7] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL https://aclanthology.org/2023.emnlp-main.183/.

[8] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

[9] Bowei He, Lihao Yin, Huiling Zhen, Shuqi Liu, Han Wu, Xiaokun Zhang, Mingxuan Yuan, and Chen Ma. Preserving llm capabilities through calibration data curation: From analysis to optimization, 2025. URL https://arxiv.org/abs/2510.10618.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

[11] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[12] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8110–8118, 2021.

[13] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. Squeezellm: dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[14] 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 llm compression and acceleration. In *MLSys*, 2024.

[15] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations, 2025. URL https://arxiv.org/abs/2405.16406.

[16] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An analysis of neural language modeling at multiple scales. *CoRR*, abs/1803.08240, 2018. doi: 10.48550/ARXIV.1803.08240. URL https://doi.org/10.48550/arXiv.1803.08240.

[17] Meta. Llama 3.1 Model Card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md, 2024.

[18] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

[19] Francesco Pio Monaco, Elia Cunegatti, Flavio Vella, and Giovanni Iacca. Frequency matters: Fast model-agnostic data curation for pruning and quantization, 2026. URL https://arxiv.org/abs/2603.16105.

[20] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[22] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

[23] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

[24] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[25] Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37:116462–116492, 2024.

[26] Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118, 2024.

[27] Miles Williams, George Chrysostomou, and Nikolaos Aletras. Self-calibration for language model quantization and pruning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10149–10167, 2025.

[28] 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 *International conference on machine learning*, pages 38087–38099. PMLR, 2023.

[29] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

[30] Qwen An Yang, Bao-Song Yang, Beichen Zhang, Bin-Yuan Hui, Bo Zheng, Bo-Wen Yu, Cheng-Yuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Hao-Ran Wei, Huan Lin, Jian Yang, Jian-Hong Tu, Jian-Wei Zhang, Jian-Xin Yang, Jia-Xin Yang, Jing-Ren Zhou, Jun-Yang Lin, Kai Dang, Ke-Ming Lu, Ke-Qin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Run-Ji Lin, Tian-Hao Li, Tingyu Xia, Xing-Zhang Ren, Xuan-Cheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhen-Ru Zhang, Zi-Han Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL https://api.semanticscholar.org/CorpusID:274859421.

[31] Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate, 2025. URL https://arxiv.org/abs/2504.19874.

[32] Zhao Zhang, Yangcheng Gao, Jicong Fan, Zhongqiu Zhao, Yi Yang, and Shuicheng Yan. Selectq: Calibration data selection for post-training quantization. *Machine Intelligence Research*, 22(3): 499–510, 2025.

[33] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A Baseline adaptation and comparison protocol

All selectors operate on the same candidate sequence IDs within a comparison. A sequence is one sample. Max-PPL ranks candidates by FP16 perplexity, Max-ActVar by aggregate activation variance, and Stratified matches the global activation-magnitude distribution. SelectQ uses activation statistics averaged over non-padding token positions, its layer-wise dynamic clustering and Knowledge Distance, and the nearest real sequence to each selected centroid. Layer scores are normalized before aggregation to avoid domination by large activation scales. ZipCal uses lexical-frequency diversity without a model-activation profile.

The fixed-pool comparison and selector-level controls use five paired calibration seeds. Candidate IDs, quantizer settings, and evaluation inputs are held fixed within each pair. No selector generates or replaces candidate data. The LLM adaptation of SelectQ should be distinguished from its original vision-model evaluation [32].

B Paired evaluation uncertainty

Table 7 reports CoverCal minus ZipCal at $K = 128$. Each of 10,000 paired bootstrap replicates resamples MMLU question IDs within each subject, preserves paired predictions, applies the same fixed subject aggregation as the harness, and averages the paired difference over the five fixed calibration seeds. For WikiText-2, replicates resample contiguous evaluation blocks, recompute paired token negative log-likelihoods and perplexities, and average over the same seed pairs. These intervals quantify evaluation-set uncertainty conditional on the observed calibration seeds, rather than population-level uncertainty over calibration seeds.

Table 7: Paired CoverCal minus ZipCal differences on LLaMA-3-8B. Accuracy differences are percentage points.

Metric	Mean paired difference	95% paired bootstrap interval
AWQ MMLU	$+0.50$	$[+0.34, +0.66]$
GPTQ MMLU	$+0.50$	$[+0.40, +0.60]$
GPTQ WikiText-2 PPL	-0.10	$[-0.12, -0.08]$

All five observed calibration-seed pairs favor CoverCal in each metric. This is descriptive: a two-sided sign test on five favorable pairs gives $p = 0.0625$, so it is not used as independent significance evidence.

C Additional model and candidate-pool evaluations

Table 8 gives FP16 reference scores for the instruction-tuned comparison. ZipCal is the strongest displayed non-CoverCal selector in all four settings. On Llama-3-8B-Instruct, Random and ZipCal both round to 62.4 MMLU, but their unrounded means are 62.38 and 62.43, respectively.

Table 8: FP16 reference scores for Table 3.

Instruction-tuned model	MMLU	IFEval
Llama-3-8B	64.0	78.1
Mistral-7B	60.4	56.2
Qwen2.5-7B	72.0	83.5
Llama-3-70B	79.6	86.1

Instruction-pool alignment. The instruction pool uses $10{,}000$ examples from the UltraChat-200k training split and each checkpoint's official chat template. The mixed pool combines $5{,}000$ instruction examples and $5{,}000$ RedPajama sequences. Table 9 separates the choice of pool from the selection rule on Llama-3-8B-Instruct under AWQ. An aligned pool helps, and CoverCal retains a gain over Random within that pool.

Table 9: Pool alignment on Llama-3-8B-Instruct, $K = 128$.

Pool and selector	MMLU	IFEval
Instruction pool, Random	62.4 ± 0.2	76.0 ± 0.4
RedPajama, CoverCal	62.7 ± 0.2	75.9 ± 0.4
50/50 mixed pool, CoverCal	63.0 ± 0.2	76.6 ± 0.3
Instruction pool, CoverCal	63.1 ± 0.2	76.8 ± 0.3

A second corpus. Table 10 uses a matched $10{,}000$-sequence C4 pool on LLaMA-3-8B with the same tokenizer, maximum length, $K = 128$, AWQ settings, and paired seeds as the RedPajama comparison. The result supports a gain on this second corpus, without establishing invariance to arbitrary domains.

Transfer of selected texts. The event universe, thresholds, masks, and weights are checkpoint-specific. Table 11 transfers texts selected using a corresponding base checkpoint to its instruction-tuned target. Transfer is partial; target-specific profiling gives the best measured MMLU in all four cases. Backend-agnostic means compatibility with AWQ and GPTQ, not model-independent selection.

Table 10: Candidate-corpus comparison on LLaMA-3-8B under AWQ.

Corpus	ZipCal MMLU	CoverCal MMLU	CoverCal weighted coverage
RedPajama	64.0 ± 0.2	64.5 ± 0.2	$84.6\% \pm 0.7\%$
C4	63.9 ± 0.2	64.3 ± 0.2	$82.7\% \pm 0.8\%$

Table 11: MMLU after transfer from corresponding base checkpoints, $K = 128$. Target backends follow Table 3.

Instruction-tuned target	Random	Base-selected texts	Target-profiled CoverCal
Llama-3-8B	62.4 ± 0.2	62.9 ± 0.2	63.1 ± 0.2
Llama-3-70B	77.9 ± 0.1	78.5 ± 0.1	78.8 ± 0.1
Mistral-7B	58.9 ± 0.2	59.3 ± 0.2	59.7 ± 0.2
Qwen2.5-7B	69.7 ± 0.2	70.0 ± 0.2	70.8 ± 0.2

D Budget efficiency

Figure 1 and Table 12 use the corrected module-level selector at every CoverCal operating point. Random is extended to $K = 1024$ to show its approach to the same quality regime. Only reported corrected operating points are plotted; unreported cells are omitted.

Table 12: Reported MMLU means underlying Figure 1. Dashes denote operating points not included in the corrected comparison.

Selector	32	64	128	256	512	1024
Random	–	–	63.1	63.8	64.1	64.4
CoverCal	63.1	64.0	64.5	64.9	–	–

E Threshold sensitivity

Tables 5 and 13 report corrected module-qualified counts, selected-set overlap, and downstream accuracy at 5σ, 6σ, and 7σ. All rows use AWQ and $K = 128$. Universe size changes with the threshold, while neighboring selected sets retain substantial overlap and MMLU varies little. This is a local sensitivity study.

For Llama-3-70B-Instruct, the additional sweep is profiling-only: event counts are 321,600, 220,480, and 139,760 at 5σ, 6σ, and 7σ; selected-set Jaccard with the 6σ set is 0.70, 1.00, and 0.66. No downstream threshold-invariance claim is made at 70B.

Table 13: Corrected module-event universe sizes and selected-set Jaccard overlap. S_j denotes selection using the $j\sigma$ threshold.

Model	$\lvert\mathcal{C}_5\rvert$	$\lvert\mathcal{C}_6\rvert$	$\lvert\mathcal{C}_7\rvert$	$J(S_5, S_6)$	$J(S_7, S_6)$
Llama-3-8B	68,410	47,025	29,640	0.73	0.69
Mistral-7B	61,080	42,130	26,910	0.71	0.67
Qwen2.5-7B	72,220	49,840	31,760	0.68	0.65
Llama-2-13B	95,840	64,920	40,610	0.72	0.68

F Computational cost

Table 14 charges each selector for its required features. Profiling is measured on one A100 for a pretokenized LLaMA-3-8B pool of 10,000 sequences, from the model pass through sparse-mask serialization. CPU greedy selection takes 8.4 seconds. Quantization and evaluation use the same backend and evaluation settings at fixed K.

Table 14: Selector preprocessing and selection time. Totals include the required features rather than assuming a free cache.

Selector	Feature extraction	Selection	Total overhead
Random	0	< 1 s	< 1 s
ZipCal	18.7 s	12.9 s	31.6 s
SelectQ	15.0 min	6.8 min	21.8 min
CoverCal	15.1 min	8.4 s	15.24 min

The totals in Table 4 are the reported rounded measurements. A reusable profile removes 15.1 minutes from a subsequent run only when checkpoint, tokenizer, pool, maximum length, and hook definition are unchanged. The one-shot comparison already charges the full profiling cost. It concerns upfront calibration time; quantized-model inference time is unchanged by the choice of selector at fixed backend.

G Module-event statistics and implementation audit

For LLaMA-3-8B, module inputs yield $32(6 \cdot 4096 + 14{,}336) = 1{,}245{,}184$ possible coordinates, including 458,752 post-SwiGLU down-projection coordinates. Table 1 distinguishes unique input masks from their copies across shared-input projections. These copies have identical binary events but can have different weights.

The layer-only implementation used one residual-stream mask per block and cannot represent the declared module universe or its module-specific squared-column weights. All coverage and threshold results in this version use the module-qualified definition. The legacy 4,950-event count and 24.8% weighted coverage are superseded, rather than being relabeled as measurements of the corrected objective.

Table 15: Selection overlap between the legacy and corrected implementations. Intersection sizes are means over five seeds; top-weight co-coverage is the fraction of top-weight-decile events covered by the corrected selection that the legacy selection also covers.

K	Shared IDs	Shared-ID fraction	Selected-set Jaccard	Top-weight co-coverage
32	20.4	63.8%	0.468 ± 0.031	$78.6\% \pm 1.7\%$
64	47.8	74.7%	0.596 ± 0.029	$85.1\% \pm 1.2\%$
128	108.0	84.4%	0.730 ± 0.026	$90.0\% \pm 0.8\%$
256	230.4	90.0%	0.818 ± 0.018	$94.3\% \pm 0.5\%$

At $K = 32/64/128$, legacy AWQ MMLU is $63.0/64.0/64.6$, while corrected MMLU is $63.1/64.0/64.5$, with standard deviations 0.2 for each reported cell. Thus module identity is necessary to define and audit the objective, but these comparisons do not establish a separate accuracy gain

from adding the module index. For Mistral-7B and Llama-2-13B, corrected event counts at 6σ are 42,130 and 64,920.

H Backend mechanism measurements and controls

For a module/channel event q, let $x_{q,t}$ range over its non-padding token activations in the selected set. Define

$$\rho_q^{(1)} = \frac{\max_t |x_{q,t}|}{\sum_t |x_{q,t}|}, \qquad \rho_q^{(2)} = \frac{\max_t x_{q,t}^2}{\sum_t x_{q,t}^2}. \tag{19}$$

These give the largest token's share of the absolute-activation sum relevant to AWQ salience and the diagonal activation energy relevant to GPTQ. Table 16 shows that the effect is concentrated in high-weight events and in upper percentiles.

Table 16: Largest-token contribution, in percent. Columns give the median, 90th percentile, and 99th percentile across the indicated events.

	$\rho^{(1)}$			$\rho^{(2)}$		
Event group	Median	90th	99th	Median	90th	99th
All covered events	0.12	1.8	9.6	0.38	7.4	41.8
Top event-weight decile	0.44	5.3	22.1	1.7	19.6	67.4

Controls. Unweighted coverage sets all event weights to one. Weight-identity permutation shuffles weights across event columns while retaining the coverage matrix and weight multiset. Unique high-weight removal excludes the highest-weight event columns uniquely supplied by the original CoverCal solution before reselection. Matched low-weight removal excludes the same number of lowest-weight columns. A length/domain-matched diversity control matches sequence length and coarse domain composition without module events. The pool, $K = 128$, backend, and paired seed are held fixed; perturbations precede selection, so selected texts may change.

Backend metrics. AWQ salience agreement is the Spearman correlation of per-channel mean absolute activations with the full-pool reference. With s denoting the final scale vector from the same AWQ search grid, and $H = AA^\top/T$ the empirical activation covariance, the relative errors are

$$E_s(\mathcal{S}) = \frac{\|s_\mathcal{S} - s_\mathcal{P}\|_2}{\|s_\mathcal{P}\|_2}, \qquad E_H(\mathcal{S}) = \frac{\|\mathrm{diag}(H_\mathcal{S}) - \mathrm{diag}(H_\mathcal{P})\|_2}{\|\mathrm{diag}(H_\mathcal{P})\|_2}. \tag{20}$$

INT4-code disagreement is the fraction of weight entries whose final integer code differs from the full-pool AWQ result. Tables 17 and 18 report mean $\pm$ standard deviation over the same five paired seeds.

Table 17: Backend measurements for selector-level controls. The full-pool reference uses identical backend settings.

Condition	AWQ salience agreement↑	AWQ scale error↓	INT4-code disagreement↓	GPTQ diagonal error↓
Random	0.66 ± 0.02	0.168 ± 0.009	$9.8\% \pm 0.5\%$	0.224 ± 0.008
Length/domain diversity	0.72 ± 0.02	0.139 ± 0.007	$8.2\% \pm 0.4\%$	0.187 ± 0.007
Unweighted coverage	0.80 ± 0.02	0.111 ± 0.006	$6.9\% \pm 0.3\%$	0.153 ± 0.006
Weights shuffled	0.70 ± 0.02	0.146 ± 0.008	$8.7\% \pm 0.4\%$	0.194 ± 0.007
High-weight removal	0.74 ± 0.02	0.132 ± 0.007	$7.9\% \pm 0.4\%$	0.181 ± 0.006
Low-weight removal	0.89 ± 0.01	0.074 ± 0.005	$4.8\% \pm 0.3\%$	0.121 ± 0.005
CoverCal	0.90 ± 0.01	0.071 ± 0.004	$4.6\% \pm 0.2\%$	0.118 ± 0.004

The final AWQ search-grid point agrees with the full-pool choice in $84.2\% \pm 1.3\%$ of modules for CoverCal, $58.4\% \pm 2.1\%$ for Random, $63.1\% \pm 2.0\%$ after weight shuffling, and $68.8\% \pm 1.8\%$ after high-weight removal. These measurements support the narrower claim that weighted event identity can affect backend calibration. They do not establish that every outlier matters or that coverage alone determines downstream accuracy.

Table 18: Downstream outcomes for the controls in Table 17, on LLaMA-3-8B at $K = 128$.

Condition	AWQ MMLU	GPTQ MMLU	GPTQ WikiText-2 PPL
Random	63.1 ± 0.3	62.5 ± 0.3	6.53 ± 0.06
Length/domain diversity	63.6 ± 0.2	63.0 ± 0.2	6.47 ± 0.05
Unweighted coverage	64.0 ± 0.2	63.6 ± 0.2	6.36 ± 0.04
Weights shuffled	63.5 ± 0.2	63.0 ± 0.2	6.48 ± 0.05
High-weight removal	63.7 ± 0.2	63.3 ± 0.2	6.43 ± 0.04
Low-weight removal	64.4 ± 0.2	64.0 ± 0.2	6.30 ± 0.03
CoverCal	64.5 ± 0.2	64.1 ± 0.2	6.28 ± 0.03

I Scope of the surrogate consistency result

The missed-tail surrogate in Section 3.4 is conditional on the residual-factor assumptions. It is not the complete Frobenius reconstruction loss, which also includes cross-channel covariance and backend-specific optimization behavior. Observing one tail token need not eliminate rounding error or represent larger future activations.

In the idealized zero-residual case, let $W_{\text{tot}} = \sum_{q \in \mathcal{C}} w_q$. The upper bound is $W_{\text{tot}} - F(\mathcal{S})$. The greedy guarantee gives

$$W_{\text{tot}} - F(\mathcal{S}_K^{\text{greedy}}) \leq W_{\text{tot}} - (1 - 1/e) F(\mathcal{S}_K^*). \tag{21}$$

This is an additive statement about the uncovered-mass bound. It is not a multiplicative $(1 - 1/e)$ approximation for minimizing reconstruction loss or the missed-tail surrogate, and it supplies no downstream-accuracy guarantee.

J Reproducibility and release information

Reproduction requires the exact model and tokenizer revisions; candidate-pool sequence IDs and preprocessing; selected indices for each selector, budget, and seed; activation hooks and mask construction; squared-column weights; AWQ/GPTQ configurations; evaluation prompts and command lines; raw per-seed results; and hardware/software versions. The selection objective, hooks, threshold construction, and greedy algorithm are specified in Sections 3 and 4. The paired bootstrap procedure is specified in Section B. Dataset and checkpoint access remains subject to their original licenses and usage terms.

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 state three deliberately scoped contributions: a module-level weighted coverage formulation for weight-only PTQ calibration, a backend-agnostic greedy selector (CoverCal) with the standard $(1-1/e)$ guarantee for the coverage objective, and empirical gains over fixed-pool baselines, including SelectQ and ZipCal, on AWQ/GPTQ INT4 for LLaMA-3-8B, with additional instruction-tuned Llama, Mistral, and Qwen models, a 70B checkpoint, and a second candidate corpus. The introduction explicitly disclaims that the contribution is a new quantizer or a downstream-accuracy bound, and Section 7 repeats this scope.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations paragraph of Section 7 discusses checkpoint-specific profiling and dependence on the candidate pool, partial transfer of selected texts, the untested architectures (mixture-of-experts, encoder-decoder, multimodal), the diagonal/tail-focused nature of the surrogate (which omits cross-channel covariance and backend optimization details), and the fact that larger random sets can approach the same quality. Section 5.3 limits the threshold claim to local stability, and Section I additionally clarifies what the surrogate consistency result does and does not claim.

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: All theoretical results (Propositions on monotonicity/submodularity and the missed-tail surrogate, and the greedy approximation Theorem) are numbered, cross-referenced, and proved in Section 3, with assumptions on the residual miss factor stated explicitly in Equations (13) and (14). The greedy guarantee invokes the classical Nemhauser–Wolsey–Fisher result, which is properly cited.

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: Sections 4 and 5 specify the activation hooks per module family, the candidate pool ($N = 10{,}000$ RedPajama sequences of maximum length 2048), the quantizer settings (AWQ/GPTQ INT4, group size 128), the threshold rule ($\mu + 6\sigma$), the budgets, the seed protocol (five paired calibration seeds), the evaluation suites and shot counts, and the greedy selector pseudocode (Algorithm 1). Section A describes the baseline adaptations. Section J enumerates the artifacts required to reproduce every reported table.

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: [Yes]

   Justification: The candidate pools (RedPajama, C4, UltraChat-200k) and all evaluation datasets are publicly available, and Section J specifies the artifacts (activation-hook code, coverage masks, selected calibration indices, AWQ/GPTQ command lines, evaluation command lines, raw evaluation JSON, hardware/software versions) that are released with the camera-ready version.

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 gives the model checkpoints, backends and group size, calibration pool size and sequence length, baseline definitions, evaluation tasks with shot counts, perplexity protocol, and seeding scheme. Threshold and weighting choices are stated in Equations (7) and (11), and Table 5 and Section E report a 5σ–7σ threshold sweep, and Table 6 reports weighting variants (unweighted coverage and shuffled weights). No retraining is performed; quantization is the only "training" step.

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: [Yes]

   Justification: The main comparison and selector-level controls are means over five paired calibration seeds, and the tables report mean $\pm$ standard deviation; MMLU standard deviations are at most 0.3, small relative to the reported gains (1.4 and 1.6 MMLU points over random at $K = 128$). Section B reports 95% paired evaluation-unit bootstrap intervals for CoverCal minus ZipCal ($[+0.34, +0.66]$ AWQ MMLU, $[+0.40, +0.60]$ GPTQ MMLU), conditional on the five observed calibration seeds.

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: Tables 4 and 14 report the compute breakdown: offline module-activation profiling takes 15.1 minutes on one NVIDIA A100 for LLaMA-3-8B with $N = 10{,}000$ pool sequences, and the sparse greedy selection step takes 8.4 seconds on CPU. Profiling can be reused across budgets and backends only when the model, tokenizer, pool, maximum length, and hook definition are unchanged.

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 work uses only publicly released model checkpoints (LLaMA-2, LLaMA-3, Mistral, Qwen2.5) and public corpora (RedPajama, C4, UltraChat-200k), does not collect new human-subject data, and does not release any new model or dataset. We have reviewed the NeurIPS Code of Ethics and the work conforms to it.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 7 notes that better calibration can preserve quality at lower deployment cost, and also flags that aggregate scores do not establish uniform preservation across domains or groups and that quantized models retain the capabilities and potential harms of their source models, recommending that practitioners evaluate quantized models on deployment-relevant subgroups rather than relying on aggregate benchmark preservation alone.

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 any new pre-trained model, generative system, or scraped dataset. It releases a calibration-selection algorithm and selected sequence indices over already-public corpora (RedPajama, C4, UltraChat-200k) for already-public model checkpoints, none of which carry a high misuse risk distinct from the underlying public artifacts.

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-2 [24], LLaMA-3 [17], Mistral-7B-v0.3 [11], Qwen2.5 [30]), the calibration corpora (RedPajama [25], C4 [21], UltraChat-200k [7]), the quantization backends (AWQ [14], GPTQ [8]), the baselines (SelectQ [32], ZipCal [19]), and the evaluation suites (MMLU [10], WikiText-2 [16], IFEval [33]) are credited via citations. Section J states that dataset and checkpoint access remains subject to their original licenses and usage terms. All assets are used within their published terms (research use of openly distributed checkpoints and datasets).

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 introduce new datasets or model checkpoints. The contribution is a selection algorithm and the resulting calibration index lists; Section J enumerates the documentation that will accompany those artifacts at release time (model and tokenizer IDs, candidate-pool sequence IDs, selected calibration indices per method/budget/backend/seed, activation-hook code, command lines, and raw evaluation JSON).

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 research with human subjects. All experiments are run on publicly released model checkpoints with publicly released text corpora and standard automated benchmarks.

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, so IRB review is not applicable.

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: [N/A]

    Justification: LLMs are the object of study in this work (the models being quantized) but are not used as an original or non-standard component of the core methodology. The selector itself is a sparse greedy weighted-coverage routine with no LLM-in-the-loop component,

and no calibration data is generated for this work; the instruction pool is drawn from the public UltraChat-200k dataset [7].