# HETEROQUANT: RESOLVING EXPERT HETEROGENEITY IN MOE QUANTIZATION VIA TARGETED CALIBRATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Mixture-of-Experts (MoE) architectures substantially expand model parameter capacity while maintaining a limited computational cost per token through sparse activation and dynamic routing mechanisms. However, during practical deployment, the complete set of expert parameters usually needs to remain in memory, and the resulting large storage overhead restricts the application of MoE models in resource-constrained environments. Post-training quantization (PTQ) is an important technique for reducing the storage and inference costs of large language models. Nevertheless, due to the significant expert heterogeneity, domain preferences, and imbalanced activation patterns in MoE models, existing quantization methods often fail to sufficiently calibrate critical experts, leading to severe performance degradation. To address these challenges, we propose ***HeteroQuant***, an effective, training-free post-training quantization framework designed to resolve expert heterogeneity through targeted calibration. Specifically, HeteroQuant achieves this by first leveraging the DSPy framework to construct a highly diverse multi-domain candidate pool, thereby comprehensively activating all expert positions to collect their global routing statistics. Based on these statistics, we implement a core two-step selection mechanism to systematically optimize the calibration data. In the first stage, we perform AlphaQ-guided structural-targeted scoring to prioritize candidate samples that precisely target and activate structurally critical experts; In the second stage, we execute activation-redundancy-aware filtering to eliminate samples with overlapping routing footprints, thereby preventing calibration waste while maximizing activation diversity. Experimental results demonstrate that under the W4A4 quantization setting of the Qwen1.5-MoE-A2.7B model, HeteroQuant achieves an average improvement of 5.09 points over the current SOTA method across eight downstream tasks.

## 1 INTRODUCTION

In recent years, large language models (LLMs) have demonstrated strong performance in various tasks, including natural language processing and code generation (Brown et al., 2020; Chen et al., 2021). Among existing mainstream model architectures, Mixture-of-Experts (MoE) models have become an important paradigm for scaling large language models. By combining sparse expert activation with dynamic routing mechanisms, MoE models can significantly increase the total parameter capacity while introducing only a limited increase in computation cost per token (Fedus et al., 2022; Jiang et al., 2024). However, although only a small number of experts are activated for each token, all expert parameters still need to remain in memory to support dynamic routing during inference (Frantar & Alistarh, 2023). For example, Qwen1.5-MoE contains approximately 14.3 billion parameters in total, while only about 2.7 billion parameters are activated for each token (Chitty-Venkata et al., 2025). Therefore, effectively compressing MoE models is critical for enabling practical deployment and efficient inference in resource-constrained environments.

Post-training quantization (PTQ) reduces model storage and memory access costs by mapping full-precision weights or activations into low-precision representations, and has achieved promising results on dense large language models (Frantar et al., 2022). However, the dynamic routing and sparse activation mechanisms of MoE models make quantization considerably more challenging. Differ-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.




Figure 1: Expert routing activation patterns across diverse domains. The top-8 code-specialized experts exhibit highly selective routing behaviors and clear domain preferences.

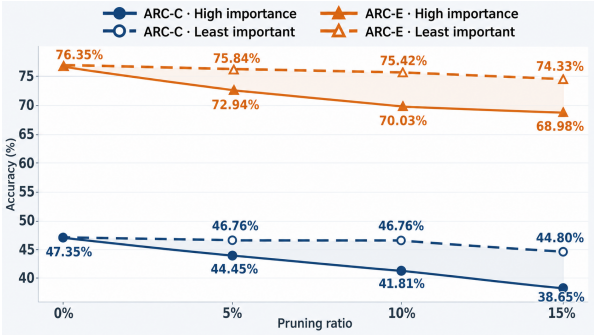


Figure 2: Accuracy comparison when pruning high-importance and least-important expert positions. Pruning critical experts causes substantially greater performance degradation.

ent experts receive significantly different numbers and distributions of calibration tokens, causing low-frequency experts and critical experts to potentially receive insufficient calibration (Hu et al., 2025; Fu et al., 2025). Moreover, different experts exhibit substantial heterogeneity in their functional roles, activation distributions, and quantization sensitivity. Existing PTQ methods generally do not explicitly consider such expert-level differences, which may lead to significant performance degradation on downstream tasks (Li et al., 2024; Hu et al., 2025; Fu et al., 2025).

To further investigate the factors affecting MoE quantization performance, we analyze two key characteristics of MoE models. First, MoE experts exhibit clear domain preferences. The same expert may show substantially different routing frequencies and activation intensities across different domains (Dong et al., 2026). As shown in Fig. 1, the top-8 experts selected based on their routing frequency in the code domain exhibit prominent spikes on code-related tokens, yet remain virtually dormant across other domains. This highly selective activation behavior poses a critical challenge for post-training quantization: if the calibration dataset lacks domain diversity, these specialized experts will remain unactivated during calibration. Consequently, their unique activation distributions cannot be accurately collected, leading to catastrophic quantization errors in specialized downstream tasks.

Second, expert contributions to MoE performance are highly imbalanced, with a small fraction of critical experts (often termed "super experts") exerting a dominant impact on downstream tasks (Su et al., 2025; Dong et al., 2026). To quantitatively verify this, we evaluate the performance of Mixtral-8×7B on ARC-E and ARC-C by progressively pruning expert positions according to our expert-importance metrics. As shown in Fig. 2, pruning just 10% of the high-importance experts triggers a catastrophic collapse in accuracy, whereas pruning the same ratio of the least-important experts yields negligible degradation. This pronounced imbalance introduces a vital constraint for MoE quantization: the quantization quality of these few critical experts largely dictates the overall post-quantization performance. Consequently, an effective calibration paradigm must move away from a superficial routing balance and instead implement targeted calibration that prioritizes the activation of these structurally and statistically critical experts.

These observations indicate that an optimal MoE calibration dataset must not only span diverse domains to fully expose routing behaviors, but also precisely target structurally critical experts while eliminating statistical redundancy to maximize calibration efficiency under a limited budget. Based on this motivation, we propose ***HeteroQuant***, a training-free post-training quantization (PTQ) framework that resolves expert heterogeneity through targeted calibration, as illustrated in Fig. 3. Specifically, HeteroQuant first leverages the DSPy framework to construct a highly diverse, multi-domain candidate pool, thereby comprehensively activating all expert positions to collect their global routing statistics. Guided by these comprehensive statistics, HeteroQuant implements a core two-step selection mechanism to systematically optimize the calibration data. In the first step, AlphaQ-guided structural-targeted scoring evaluates candidate samples based on expert structural sensitivity, ensuring that prompts precisely targeting and activating structurally critical experts are prioritized.

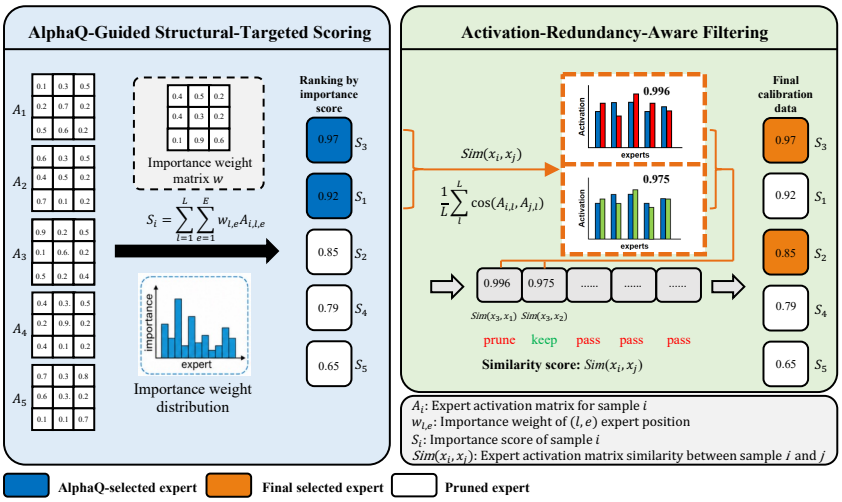


Figure 3: Overview of the expert-aware calibration sample selection pipeline. Candidates are ranked by AlphaQ-guided importance scores and then filtered according to the similarity of their routing patterns, removing redundant samples until the target calibration-set size is reached.

In the second step, activation-redundancy-aware filtering sequentially evaluates these prioritized candidates, filtering out those with overlapping routing footprints to eliminate statistical redundancy and maximize calibration diversity. Experimental results demonstrate that this targeted, redundancy-free calibration paradigm consistently outperforms both conventional datasets and routing-balanced baselines, successfully preserving MoE capabilities under aggressive low-bit settings without any parameter-tuning overhead. The main contributions of this work are summarized as follows:

- We propose ***HeteroQuant***, a training-free post-training quantization framework, and introduce AlphaQ-guided structural-targeted scoring to resolve expert heterogeneity by prioritizing calibration samples that precisely activate structurally critical experts based on parameter sensitivity.
- We propose activation-redundancy-aware filtering as a key statistical refinement strategy, which compares layer-wise routing footprints using local normalization to eliminate overlapping activation patterns and maximize calibration diversity under a limited budget.
- We conduct extensive experiments on three mainstream MoE models across seven datasets, demonstrating that HeteroQuant consistently outperforms existing SOTA methods (e.g., achieving a 11.30 percentage-point improvement on GSM8K) while maintaining extreme efficiency.

## 2 Related Work

**MoE Quantization:** Post-training quantization (PTQ) effectively reduces memory and inference overheads for large language models (Frantar et al., 2022). For sparse MoEs, existing methods focus on weight-only compression (Frantar & Alistarh, 2023), mixed-precision allocation based on expert sensitivity or activation (Huang et al., 2025), and expert-aware activation quantization (Fu et al., 2025). However, these algorithms typically treat calibration datasets as static inputs, remaining highly sensitive to the underlying data distribution.

**Calibration for MoE Quantization:** Calibrating MoEs is challenging due to the highly imbalanced expert activation caused by sparse routing. To improve calibration coverage, MoEQuant (Hu et al., 2025) introduces expert-balanced sampling, while DynaMo (Zheng et al., 2026) models routing dynamics under distribution shifts. Alternatively, AlphaQ (Yang et al., 2026) bypasses calibration during bit allocation by deriving expert importance from static weight spectra, though representative data remains indispensable for scale estimation.

**Heavy-Tailed Self-Regularization:** Heavy-Tailed Self-Regularization (HT-SR) evaluates neural network generalization by analyzing the empirical spectral density of weight correlation matrices (Mahoney & Martin, 2019; Martin et al., 2021), where heavier-tailed spectra signify stronger learned representations (Zhou et al., 2023). AlphaQ (Yang et al., 2026) leverages this theory to estimate offline expert importance directly from parameter weights without calibration overhead.

Unlike existing approaches that treat calibration data as a static given, HeteroQuant optimizes the calibration data engine itself, making it seamlessly compatible with any PTQ pipeline. While prior strategies either blindly pursue flat routing balance (*e.g.*, EBSS) or ignore data-level optimization (*e.g.*, AlphaQ), HeteroQuant reconciles expert structural importance with routing diversity. Rather than using HT-SR as a static, offline diagnostic tool, we dynamically map parameter spectral sensitivities to token-level routing probabilities, producing a highly targeted, redundancy-free calibration dataset under a strict budget.

# 3 HETEROQUANT

In this section, we present the systematic design of HeteroQuant to resolve expert heterogeneity in MoE models. We first establish the mathematical foundations of post-training quantization in preliminaries. We then detail our expert-aware calibration scheme: beginning with DSPy-driven calibration pool generation to thoroughly expose global routing behaviors, followed by AlphaQ-guided calibration set selection to target structurally critical experts, and concluding with activation-redundancy-aware selection to eliminate statistical overlap and maximize routing diversity.

## 3.1 PRELIMINARIES

**Post-Training Quantization:** Post-training quantization (PTQ) converts high-precision model parameters into low-bit representations to reduce memory and computation overhead during inference (Frantar et al., 2022). For a floating-point weight matrix $W$, the quantization process is:

$$q = \text{clip}\left(\text{round}\left(W/s\right), q_{\min}, q_{\max}\right), \tag{1}$$

where $q$ is the quantized integer, $s$ is the scale, and $q_{\min}, q_{\max}$ are the bit-width limits. The quantized weight is recovered via $\hat{W} = s \cdot q$. The scale $s$ directly dictates the quantization error and the resulting PTQ performance.

**Calibration Data:** Calibration data are representative samples used to collect activation distributions $X$ via forward propagation to determine quantization parameters (Frantar et al., 2022; Li et al., 2024). For symmetric uniform quantization, the scale is computed as $s = \max(|X|)/(2^{b-1} - 1)$, where $b$ is the quantization bit width. As $s$ is entirely determined by the calibration statistics, the quality of this dataset directly dictates model performance.

**Challenge Analysis:** Existing MoE quantization typically relies on domain-specific datasets (e.g., WikiText-2 (Merity et al., 2016)), which triggers severe activation imbalance: only domain-associated experts are activated, while others remain dormant in a long-tailed distribution (Dong et al., 2026). Although EBSS (Hu et al., 2025) balances expert activations, it overlooks the non-uniform structural importance of different expert positions (Su et al., 2025). Furthermore, selecting samples solely based on individual importance scores leads to routing redundancy, wasting the limited calibration budget.

Based on these insights and our pruning experiments, we define sufficient activation of important expert positions and diversity of routing patterns as our primary objectives, formulated through two core challenges:

- **Challenge 1:** How to generate calibration data that sufficiently exposes the activation patterns of expert positions across different domains, thereby supporting accurate expert-importance estimation.
- **Challenge 2:** How to select samples that strongly activate important experts while avoiding redundant expert routing patterns.

To address these challenges, we propose HeteroQuant, a training-free framework that constructs a diverse multi-domain candidate pool to address Challenge 1, followed by a core two-step selection mechanism to resolve Challenge 2.

### 3.2 DSPy-Based Calibration Pool Generation

To address Challenge 1 and construct a calibration dataset that thoroughly exposes the domain-specific routing behaviors of heterogeneous experts, we first generate a highly diverse multi-domain candidate pool.

In this work, we leverage DSPy (Khattab et al., 2023), a declarative programming framework for large language models that decomposes complex tasks into composable reasoning modules and automatically optimizes prompt templates, to invoke the DeepSeek API. This process automatically constructs high-quality prompts covering a wide spectrum of knowledge domains, which are subsequently fed into the target MoE model to generate candidate calibration sequences. The structured prompt generation process consists of three components: Signature, Module, and Optimizer.

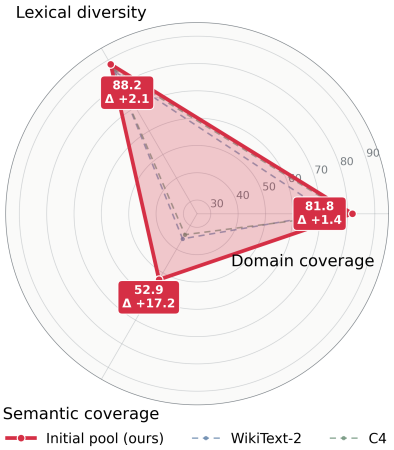


Figure 4: **Comparison of domain richness across calibration pools.** HeteroQuant's initial pool shows substantially higher domain richness than WikiText-2 and C4.

First, the Signature defines the inputs and outputs of the prompt generation task. For the $i$-th candidate prompt, its input conditions are represented as:

$$\mathbf{z}_i = (g_i, d_i, r_i, l_i, c), \tag{2}$$

where $g_i$ denotes the high-level domain category, $d_i$ denotes the specific knowledge domain, $r_i$ denotes the target capability or reasoning pattern, $l_i$ denotes the difficulty level, and $c$ denotes the domain coverage objective. The output is the prompt $p_i$ used to generate calibration text.

Subsequently, we employ the Chain-of-Thought Module in DSPy to instantiate the above Signature and generate prompts, with DeepSeek serving as the underlying language model. The prompt generation process is formulated as

$$p_i = \mathcal{M}_{\mathrm{CoT}}\left(\Sigma, \mathbf{z}_i; \theta\right), \tag{3}$$

where $\Sigma$ denotes the task Signature and $\theta$ denotes the parameters of the DeepSeek model. This module jointly considers constraints related to domains, capabilities, and difficulty levels, thereby improving the domain diversity and quality of the generated prompts.

Finally, the Optimizer improves the prompt distribution through domain-diverse sampling, difficulty allocation, and randomization strategies. We divide the candidate data into multiple major domains and attempt to maintain a relatively balanced number of generated samples across different domains, satisfying

$$n_k \approx \frac{N}{K}, \tag{4}$$

where $N$ denotes the total number of prompts, $K$ denotes the number of domain categories, and $n_k$ denotes the number of prompts belonging to the $k$-th domain. The generated prompts are further fed into the model to be quantized to obtain candidate calibration samples:

$$x_i \sim P_\phi(x \mid p_i), \tag{5}$$

which finally constitute the multi-domain initial calibration pool:

$$\mathcal{D}_{\mathrm{init}} = \{x_1, x_2, \ldots, x_M\}. \tag{6}$$

As shown in Fig. 4, we compare the initial data pool with conventional calibration datasets such as WikiText-2 (Merity et al., 2016) and C4 (Raffel et al., 2020). The domain richness of the initial data pool is evaluated from three dimensions: topic entropy, average semantic cosine distance, and domain coverage. The results demonstrate that the generated initial data pool exhibits high domain diversity.

### 3.3 AlphaQ-Guided Structural-Targeted Scoring

While the DSPy-generated candidate pool $\mathcal{D}_{\text{init}}$ successfully activates the entire MoE model, selecting the final calibration set from this vast pool brings us to our second challenge: balancing expert importance with routing redundancy. Because MoE experts are functionally unequal, treating every expert position with equal weight is a major pitfall. To tackle the first aspect of this second challenge, we introduce our first selection stage: AlphaQ-guided structural-targeted scoring. This mechanism leverages expert parameter sensitivity to identify structurally critical positions, scoring and prioritizing candidate samples based on how precisely they target and trigger these vital experts.

We first employ the AlphaQ index (Yang et al., 2026) to estimate the structural sensitivity and importance of each expert position. For the $e$-th expert in the $l$-th MoE layer, we compute the AlphaQ values of its gate, up, and down projection weight matrices, and use their average as the AlphaQ value of the corresponding expert position:

$$\alpha_{l,e} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \text{AlphaQ}\left(W_{l,e}^{(p)}\right). \tag{7}$$

Here, $\mathcal{P} = \{\text{gate}, \text{up}, \text{down}\}$ denotes the set of projection modules, and $W_{l,e}^{(p)}$ denotes the weight matrix of projection $p$ for expert $e$ in layer $l$. According to AlphaQ (Yang et al., 2026), a smaller $\alpha_{l,e}$ indicates a structurally more sensitive and critical expert position. Since the subsequent sample scoring procedure requires weights that are positively correlated with expert importance, we transform the AlphaQ values into positive expert-position importance weights:

$$\widetilde{\omega}_{l,e} = \left(\frac{\alpha_{\text{med}}}{\alpha_{l,e} + \varepsilon}\right)^{\gamma}, \tag{8}$$

where $\alpha_{\text{med}}$ denotes the median AlphaQ value across all expert positions, $\varepsilon$ is a small constant introduced to prevent numerical instability, and $\gamma$ is a scaling coefficient controlling the weight differences among experts. The weights of all expert positions are then normalized as:

$$\omega_{l,e} = \frac{\widetilde{\omega}_{l,e}}{\sum_{l'=1}^{L} \sum_{e'=1}^{E} \widetilde{\omega}_{l',e'}}, \tag{9}$$

where $L$ denotes the number of MoE layers and $E$ denotes the number of experts in each layer. After this transformation, structurally critical expert positions receive larger importance weights.

Next, each candidate sample $x_i$ from the initial pool $\mathcal{D}_{\text{init}}$ is fed into the MoE model for forward propagation to collect its empirical routing behavior. For a candidate sample $x_i$, its cumulative routing probability at expert position $(l, e)$ is defined as:

$$\widetilde{A}_{i,l,e} = \sum_{t=1}^{T_i} \mathbb{I}\left[e \in \text{TopK}\left(\mathbf{r}_{i,l,t}\right)\right] p_{i,l,t,e}, \tag{10}$$

where $T_i$ denotes the number of tokens in sample $x_i$, $\mathbf{r}_{i,l,t}$ denotes the routing scores produced by the router in layer $l$ for the $t$-th token, $p_{i,l,t,e}$ denotes the routing probability of expert $e$ after Softmax, and $\mathbb{I}[\cdot]$ is the indicator function.

To eliminate the influence of differences in sample length and overall activation magnitude, the expert activation matrix of each sample is globally normalized as:

$$A_{i,l,e} = \frac{\widetilde{A}_{i,l,e}}{\sum_{l'=1}^{L} \sum_{e'=1}^{E} \widetilde{A}_{i,l',e'}}. \tag{11}$$

The normalized value $A_{i,l,e}$ represents the relative activation proportion of candidate sample $x_i$ at expert position $(l, e)$.

Combining the structural expert-position importance weights and sample routing footprints, the AlphaQ-guided structural-targeted score of candidate sample $x_i$ is formulated as:

$$S_i = \sum_{l=1}^{L} \sum_{e=1}^{E} \omega_{l,e} A_{i,l,e}. \tag{12}$$

This score measures the extent to which a sample targets and activates structurally critical expert positions. A candidate that strongly activates experts with larger structural weights obtains a correspondingly higher score. As illustrated in the first phase of the selection pipeline in Fig. 3, all candidates are sorted in descending order according to their importance scores $S_i$, serving as the prioritized queue for subsequent redundancy filtering.

### 3.4 Activation-Redundancy-Aware Filtering

While the scoring mechanism in the first stage successfully highlights samples that target critical experts, selecting calibration data solely based on these raw scores inevitably leads to a statistical trap. Because high-scoring prompts are naturally drawn to the same set of dominant experts, greedily selecting the top-ranked candidates will cause severe clustering in a few routing paths. Under a strict budget of $|\mathcal{D}_{\text{cal}}| = 128$, this statistical overlap wastes precious calibration slots and starves other experts. To fully address the Challenge 2, we introduce activation-redundancy-aware filtering.

To compare the routing footprints of different samples objectively, we must isolate their activation patterns from their absolute activation scales, which vary heavily across different layers and sequence lengths. To achieve this, we locally normalize the expert activation distribution of sample $x_i$ within each layer $l$:

$$\bar{A}_{i,l,e} = \frac{A_{i,l,e}}{\sum_{e'=1}^{E} A_{i,l,e'} + \varepsilon}, \tag{13}$$

where $\bar{A}_{i,l,e}$ represents the relative activation proportion of expert $e$ within layer $l$, and $\varepsilon$ prevents division by zero. This local normalization ensures that each layer contributes equally to the similarity measurement, regardless of its global activation magnitude. The structural similarity between the routing footprints of sample $x_i$ and $x_j$ is then measured by the mean cosine similarity of their relative activation vectors across all MoE layers:

$$\text{Sim}(x_i, x_j) = \frac{1}{L} \sum_{l=1}^{L} \cos\left(\bar{\mathbf{A}}_{i,l}, \bar{\mathbf{A}}_{j,l}\right), \tag{14}$$

where $\bar{\mathbf{A}}_{i,l} \in \mathbb{R}^E$ denotes the normalized expert activation vector of sample $x_i$ in layer $l$. A larger similarity value indicates a higher degree of overlap in their layer-wise expert selection.

With these formulations established, we perform a redundancy-free greedy sweep. We initialize the final calibration set $\mathcal{D}_{\text{cal}}$ with the highest-scoring candidate from our sorted queue, and then sequentially evaluate the remaining candidates $x_i$. For each candidate, we compute its peak similarity against the already accepted set:

$$R_i = \max_{x_j \in \mathcal{D}_{\text{cal}}} \text{Sim}(x_i, x_j). \tag{15}$$

We conditionally admit the candidate to our calibration set if and only if it introduces sufficiently novel routing profiles, verified by:

$$R_i < \tau, \tag{16}$$

where $\tau$ is the similarity threshold. If $R_i \geq \tau$, the sample is skipped as redundant. This scanning process continues down the prioritized queue until our budget limit is reached:

$$|\mathcal{D}_{\text{cal}}| = 128. \tag{17}$$

By keeping the calibration size at 128, we strictly align with the standard WikiText-2 settings used in conventional LLM quantization baselines, ensuring a fair and rigorous comparison under the exact

Table 1: Main results under W4A4R16. PPL is reported on WikiText-2 and C4; accuracy (%) is reported on eight downstream benchmarks. Avg. denotes mean accuracy, and the best quantized results are bolded.

| Model | Method | PPL ↓ | | Accuracy (%) ↑ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WikiText-2 | C4 | ARC-C | ARC-E | BoolQ | HS | PIQA | WG | GSM8K | C-Eval | Avg. |
| OLMoE-1B-7B | *FP16* | 9.06 | 15.08 | 49.49 | 78.45 | 76.45 | 80.68 | 79.00 | 69.22 | 67.85 | 31.35 | 66.56 |
| | QuaRot | 10.45 | 19.18 | 45.73 | 71.42 | **75.14** | 78.27 | **75.57** | 64.56 | 55.65 | 31.63 | 62.25 |
| | EAQuant | 11.02 | 16.89 | 46.08 | 72.85 | 74.46 | **78.50** | 74.05 | 64.72 | 57.47 | 28.32 | 62.06 |
| | **HeteroQuant** | **10.33** | **14.98** | **47.18** | **73.48** | 74.65 | 78.44 | 75.19 | **66.06** | **58.76** | **32.76** | **63.32** |
| Qwen1.5-MoE-A2.7B | *FP16* | 6.79 | 9.70 | 44.20 | 69.36 | 79.82 | 77.23 | 80.47 | 68.98 | 60.88 | 81.35 | 70.29 |
| | QuaRot | 7.87 | 11.41 | 40.10 | 64.56 | 73.52 | 73.44 | 77.42 | 64.80 | 29.34 | 65.08 | 61.03 |
| | EAQuant | 8.21 | 10.58 | 39.16 | 62.84 | 75.23 | 71.50 | 77.42 | 63.69 | 19.71 | 64.41 | 59.25 |
| | **HeteroQuant** | **7.61** | **9.69** | **40.19** | **66.54** | **75.75** | **74.40** | **78.29** | **65.82** | **40.64** | **71.47** | **64.14** |
| Mixtral-8×7B | *FP16* | 3.84 | 6.87 | 59.81 | 83.29 | 85.23 | 85.95 | 85.20 | 75.85 | 65.88 | 51.04 | 74.03 |
| | QuaRot | 5.26 | 7.45 | 53.41 | 78.87 | 81.28 | 80.27 | 80.74 | 69.14 | 36.85 | 36.93 | 64.69 |
| | EAQuant | **4.10** | 6.97 | 56.48 | 81.69 | **84.65** | 81.96 | 81.18 | 74.19 | 46.70 | 41.23 | 68.51 |
| | MoEQuant | 4.23 | 7.09 | 55.46 | 81.73 | 83.15 | **82.48** | 81.61 | 73.65 | 46.70 | **43.16** | 68.49 |
| | **HeteroQuant** | 4.13 | **6.96** | **57.34** | **82.58** | 83.64 | 82.25 | **82.54** | **74.74** | **47.01** | 43.02 | **69.14** |

same computational budget. Ultimately, this two-step selection paradigm elegantly reconciles the tension between expert importance and routing diversity—ensuring that our calibration data deeply exercises structurally critical parameter regions without wasting resources on repetitive routing pathways.

# 4 Experiments

## 4.1 Experimental Setup

**Models and Baselines:** We evaluate HeteroQuant on three publicly available sparse MoE language models: OLMoE-1B-7B-0125-Instruct, Qwen1.5-MoE-A2.7B, and Mixtral-8×7B-v0.1 (Muennighoff et al., 2025; Qwen Team, 2024; Jiang et al., 2024). We compare HeteroQuant against the FP16 model and representative W4A4 post-training quantization methods, including QuaRot, DuQuant, MoEQuant, and EAQuant. For each baseline, we follow its released implementation and recommended calibration protocol whenever available. We evaluate router precision at 16-bit (R16) and 8-bit (R8): HeteroQuant-R16 is compared with QuaRot-R16, MoEQuant-R16, and EAQuant-R16, while HeteroQuant-R8 is compared with DuQuant-R8 and EAQuant-R8.

**Quantization and Implementation Details:** We use W4A4 quantization with asymmetric per-channel weights, dynamic per-token activations, block size 128, and expert-aware smoothing ($\alpha = 0.6$), merging expert projection scales by the channel-wise maximum.Experiments are implemented with PyTorch and Transformers on a GPU with 96 GB of memory; W4A4R8 results are provided in Appendix A.

**Calibration Configurations:** While the baseline methods rely on the standard 128 WikiText-2 calibration sequences, HeteroQuant first constructs a larger multi-domain initial pool $\mathcal{D}_{\text{init}}$ containing 512 candidate sequences (each with 4,096 tokens). Specifically, we employ a DSPy-driven generation framework with the DeepSeek model to automatically construct diverse prompts covering multiple knowledge domains, reasoning capabilities, and difficulty levels. These prompts are then fed into the model to be quantized to generate candidate calibration samples, forming the initial calibration pool with broader domain coverage and richer activation patterns. From this initial pool, we extract the final 128 calibration sequences using our expert-aware selection strategy: candidates are first prioritized via their AlphaQ-guided importance scores (based on PL Alpha Hill spectral weights), and then greedily filtered using a layer-wise cosine similarity threshold to remove redundant routing patterns.

**Datasets:** We evaluate downstream performance via the LM Evaluation Harness (Gao et al., 2023) across eight diverse benchmarks: ARC-Easy, ARC-Challenge, BoolQ, HellaSwag, PIQA, WinoGrande, C-Eval (Valid), and GSM8K (Clark et al., 2018; Cobbe et al., 2021). We report standard task-specific metrics and compute the aggregate downstream performance as the unweighted macro-average of these task scores. Additionally, we report language modeling perplexity (PPL) on the WikiText-2 test set and C4 validation set using non-overlapping 4,096-token windows.

Table 2: Ablation study of HeteroQuant on OLMoE-1B-7B (W4A4R16). DSPy, AlphaQ, and Filtering denote DSPy-based calibration pool generation, AlphaQ-guided structural-targeted scoring, and activation-redundancy-aware filtering, respectively. The last row denotes the complete HeteroQuant. We report perplexity on WikiText-2 (WT2) and C4 and average accuracy across six zero-shot downstream tasks (excluding GSM8K and C-Eval).

| DSPy | AlphaQ | Filtering | WT2 PPL ↓ | C4 PPL ↓ | Avg. ↑ |
|---|---|---|---|---|---|
| × | ✓ | ✓ | 10.20 | 16.01 | 68.68 |
| ✓ | × | ✓ | 10.34 | 16.00 | 68.12 |
| ✓ | ✓ | × | **10.18** | 15.92 | 68.69 |
| ✓ | ✓ | ✓ | 10.33 | **14.98** | **69.17** |

### 4.2 Main Results

Table 1 compares HeteroQuant with representative MoE quantization methods across three architectures in terms of perplexity and downstream task performance.

Overall, HeteroQuant achieves the best performance among all evaluated quantization methods. On the Qwen1.5-MoE-A2.7B model, HeteroQuant consistently outperforms QuaRot and EAQuant across all reported benchmarks. Specifically, compared with EAQuant, HeteroQuant improves the performance on ARC-C, ARC-E, BoolQ, HellaSwag, PIQA, and WinoGrande by 1.03, 3.70, 0.52, 2.90, 0.87, and 2.13 percentage points, respectively. These results demonstrate that improving calibration quality is crucial for preserving the capability of quantized MoE models, especially under aggressive low-bit quantization settings.

Furthermore, HeteroQuant maintains strong generalization across different MoE architectures. Compared with existing quantization approaches, HeteroQuant achieves lower perplexity on both WikiText-2 and C4 while obtaining higher downstream accuracy, indicating that the proposed expert-aware calibration strategy better captures the activation characteristics of heterogeneous experts. Unlike conventional calibration methods that treat all experts equally, HeteroQuant explicitly prioritizes structurally important experts and selects calibration samples with diverse routing patterns, thereby reducing calibration bias caused by sparse and imbalanced expert activation.

### 4.3 Ablation Study

Table 2 evaluates the contribution of each calibration component in terms of both language-modeling perplexity and average downstream accuracy. Removing the multi-domain candidate pool decreases the average accuracy from 69.17% to 68.68%. This result indicates that a domain-diverse candidate pool is important for exposing heterogeneous expert-routing behaviors and constructing representative calibration data.

Replacing AlphaQ-guided importance scoring with random candidate selection causes the largest average accuracy degradation, reducing it by 1.05 percentage points to 68.12%. This demonstrates that routing diversity alone is insufficient: calibration samples must also preferentially activate structurally critical expert positions.

Removing redundancy filtering reduces the average accuracy to 68.69%. This suggests that importance-based ranking alone may select samples with highly overlapping routing footprints, limiting the effective diversity of the calibration set. Overall, combining multi-domain pool construction, importance-guided scoring, and redundancy-aware filtering achieves the best aggregate downstream accuracy while maintaining favorable perplexity on both WikiText-2 and C4.

## 5 Conclusion

We propose HeteroQuant, a training-free MoE PTQ framework that improves calibration through a DSPy-generated multi-domain pool, AlphaQ-guided structural scoring, and activation-redundancy-aware filtering. Experiments across multiple MoE models show that HeteroQuant consistently improves quantized performance, demonstrating the effectiveness of expert-aware calibration.

## AI USE STATEMENT

We used large language models to assist with improving the grammar and wording of the manuscript. In addition, as described in the paper, DeepSeek is used within the DSPy-based pipeline to generate prompts for constructing the initial calibration pool. All technical content, experimental design, analysis, and conclusions were developed and verified by the authors.

## ETHICS STATEMENT

This study evaluates post-training quantization methods on publicly available MoE models and benchmark datasets. The experiments involve no human participants and do not collect or use personal or sensitive data. HeteroQuant focuses on improving the efficiency of model quantization and does not introduce additional model capabilities beyond those of the underlying models.

## REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Krishna Teja Chitty-Venkata, Sylvia Howland, Golara Azar, Daria Soboleva, Natalia Vassilieva, Siddhisanket Raskar, Murali Emani, and Venkatram Vishwanath. Moe-inference-bench: Performance evaluation of mixture of expert large language and vision models. In *Proceedings of the SC'25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1502–1511, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Zican Dong, Han Peng, Peiyu Liu, Xin Zhao, Dong Wu, Feng Xiao, and Zhifeng Wang. Domain-specific pruning of large mixture-of-experts models with few-shot demonstrations. *Advances in Neural Information Processing Systems*, 38:103552–103577, 2026.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Elias Frantar and Dan Alistarh. Qmoe: Practical sub-1-bit compression of trillion-parameter models. *arXiv preprint arXiv:2310.16795*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Zhongqian Fu, Tianyi Zhao, Ning Ding, Xianzhi Yu, Xiaosong Li, Yehui Tang, and Yunhe Wang. Eaquant: Enhancing post-training quantization for moe models via expert-aware optimization. *arXiv preprint arXiv:2506.13329*, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework

for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. *arXiv preprint arXiv:2505.03804*, 2025.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts llms gains more, 2025. URL https://arxiv.org/abs/2410.06270.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Pingzhi Li, Xiaolong Jin, Yu Cheng, and Tianlong Chen. Examining post-training quantization for mixture-of-experts: A benchmark. 2024.

Michael Mahoney and Charles Martin. Traditional and heavy tailed self regularization in neural network models. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4284–4293. PMLR, 2019. URL https://proceedings.mlr.press/v97/mahoney19a.html.

Charles H. Martin, Tongsu Peng, and Michael W. Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12 (1):4122, 2021. doi: 10.1038/s41467-021-24025-8. URL https://doi.org/10.1038/s41467-021-24025-8.

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

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2025. URL https://arxiv.org/abs/2409.02060.

Qwen Team. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters. https://qwenlm.github.io/blog/qwen-moe/, March 2024.

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.

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

Wanqi Yang, Yuexiao Ma, Alexander Conzelmann, Xiawu Zheng, Michael W Mahoney, T Konstantin Rusch, and Shiwei Liu. Alphaq: Calibration-free bit allocation for mixture-of-experts quantization. *arXiv preprint arXiv:2606.04980*, 2026.

Zihao Zheng, Xiuping Cui, Size Zheng, Maoliang Li, Jiayu Chen, Yun Liang, and Xiang Chen. Dynamo: Runtime switchable quantization for moe with cross-dataset adaptation, 2026. URL https://arxiv.org/abs/2503.21135.

Yefan Zhou, Tianyu Pang, Keqin Liu, Charles H. Martin, Michael W. Mahoney, and Yaoqing Yang. Temperature balancing, layer-wise weight analysis, and neural network training. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63542–63572, 2023.

## A Representative Examples of the Initial Calibration Pool

To further illustrate the diversity and quality of the initial calibration pool generated by HeteroQuant, we provide representative examples selected from different domain groups in Table 3. The initial pool is generated through our DSPy-driven framework and contains samples covering various knowledge domains. These examples demonstrate that the generated calibration data contain diverse semantic structures, reasoning patterns, and knowledge areas, which helps expose different expert routing behaviors before the importance-aware selection stage.

Table 3: Representative examples from the initial calibration pool generated by HeteroQuant. The selected excerpts span seven domain groups and demonstrate the semantic diversity of the generated calibration samples.

| **Domain** | **Topic** | **Representative excerpt** |
|---|---|---|
| Engineering Technology | Extreme-environment thermal management | The output provides insights into how various thermal management strategies might perform under extreme conditions. Passive cooling, active cooling, advanced materials, and redundant system design are discussed to improve reliability under high temperature, low temperature, radiation, and pressure constraints. |
| Computing AI | Scaling zero-shot learning | Zero-shot learning aims to recognize objects in unseen classes given only image descriptions or attributes. Scaling such systems introduces challenges including computational complexity, data annotation quality, and resource constraints. |
| Social Business Policy | Mechanism design in globalized economies | Mechanism design plays an important role in shaping economic institutions by considering participant preferences and constraints. It improves efficiency, innovation, and fairness in interconnected markets. |
| Humanities Arts | Cultural knowledge and diversity | Cultural knowledge provides values, norms, and perspectives that guide human understanding. Cultural diversity enriches society by promoting creativity, tolerance, and interaction among different communities. |
| Environment Energy | Environmental policy impact | Historical environmental agreements have influenced contemporary policies through global cooperation, legal frameworks, and technological advancements for sustainability. |
| Mathematics Physics | Probability theory and statistical inference | Probability theory provides mathematical tools for modeling uncertainty, updating beliefs, and supporting statistical inference from observed data. |
| Life Health | Ethical implications of tissue engineering | Tissue engineering combines biology, chemistry, materials science, and engineering to develop medical applications while raising ethical, economic, and regulatory considerations. |

## B Additional Results under Reduced Router Precision

In addition to the main W4A4R16 experiments presented in the paper, we further evaluate HeteroQuant under the W4A4R8 setting on OLMoE-1B-7B-0125-Instruct. In this setting, expert linear layers are quantized to 4-bit weights and 4-bit activations, while the router is quantized to 8-bit precision. This experiment investigates whether the proposed expert-aware calibration strategy remains effective when the router precision is reduced.

As shown in Table 4, HeteroQuant achieves an average downstream accuracy of 63.93%, outperforming EAQuant and DuQuant by 2.35 and 8.68 percentage points, respectively. In particular, HeteroQuant achieves 64.37% accuracy on GSM8K, which is 7.05 percentage points higher than EAQuant.

Furthermore, HeteroQuant achieves the best quantized performance on ARC-C, ARC-E, BoolQ, WinoGrande, GSM8K, and C-Eval. These results demonstrate that the proposed calibration strategy

Table 4: Performance comparison under the W4A4R8 quantization setting on OLMoE-1B-7B-0125-Instruct. Expert layers use 4-bit weights and activations, while the router uses 8-bit precision. The best quantized results are highlighted in bold.

| Model | Method | PPL ↓ | | Accuracy (%) ↑ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | WikiText-2 | C4 | ARC-C | ARC-E | BoolQ | HS | PIQA | WG | GSM8K | C-Eval | Avg. |
| OLMoE-1B-7B | *FP16* | 9.06 | 15.08 | 49.49 | 78.45 | 76.45 | 80.68 | 79.00 | 69.22 | 67.85 | 31.43 | 66.57 |
| | DuQuant | 10.44 | 16.09 | 44.96 | 73.11 | 74.58 | **78.66** | **75.51** | 64.48 | 0.68 | 30.01 | 55.25 |
| | EAQuant | 10.98 | 16.78 | 45.65 | 72.18 | 72.51 | 78.49 | 73.83 | 64.01 | 57.32 | 28.68 | 61.58 |
| | **HeteroQuant** | **10.19** | **14.91** | **48.21** | **73.44** | **74.89** | 78.25 | 74.92 | **65.75** | **64.37** | **31.58** | **63.93** |

remains effective under reduced router precision. By prioritizing samples that activate structurally important experts and eliminating redundant routing patterns, HeteroQuant provides robust calibration statistics even when both expert modules and router parameters are quantized.