# Less is MoE: Trimming Experts in Domain-Specialist Language Models

**Haoze He**[1*] **Xinkai Zou**[2*] **Xuan Jiang**[3] **Xingyuan Ding**[1]
**Ao Qu**[3] **Juncheng Billy Li**[1] **Heather Miller**[1]

[1]Carnegie Mellon University [2]UCSD [3]MIT
{haozeh, xingyuad, junchenl, heather.miller}@cs.cmu.edu
jayzou@ucsd.edu {xuanj, qua}@mit.edu

**Project Page:** https://less-is-moe.github.io/
**Code Repo:** https://github.com/HectorHHZ/Less-is-MoE

## Abstract

Mixture-of-Experts (MoE) models achieve strong performance through conditional computation, but their large parameter footprint poses deployment challenges. Prior MoE compression approaches catastrophically fail when evaluated on general-purpose benchmarks beyond commonsense reasoning. We trace this failure to the granularity of compression: important capabilities are distributed across experts but concentrated in FFN sparse intermediate dimensions. To identify these dimensions, we use Fisher importance which outperforms activation-, router-score-, and magnitude-based alternatives, and identifies tiny sets of task-critical dimensions: in Qwen1.5-MoE, removing as few as 12 of 2.03M routed-FFN intermediate dimensions collapses GSM8K accuracy while largely preserving factual-knowledge performance. Building on this, we propose **Fisher-MoE**, which operates within FFN to remove intermediate dimensions ranked by Fisher importance. At the same 50% MoE compression ratio, Fisher-MoE preserves model capability, while reducing weight memory by $\sim$45% and improving inference throughput by 21%. These findings suggest intermediate dimension granularity is an effective unit for both compression and ranking where capability concentrates in MoE models. Our implementation is available in an anonymous repository.

## 1 Introduction

Mixture-of-Experts (MoE) models have emerged as a dominant paradigm for scaling language model capacity while maintaining efficient inference through conditional computation (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). This enables models with tens of billions of parameters to achieve efficient inference, but their large total parameter footprint still poses significant challenges for deployment in terms of memory, storage, and serving.

To reduce this footprint, prior work compresses MoE models by removing or merging experts based on heuristic importance metrics, such as activation frequency (Muzio et al., 2024; Lu et al., 2024a; Chen et al., 2022), router scores (Xie et al., 2024; Gu et al., 2025), or weight magnitudes (Lee et al., 2025; Yang et al., 2024b; Li et al., 2023; Chen et al., 2025). Despite differences in these metrics, existing approaches share a common design choice: compression is performed at the granularity of entire experts. Moreover, existing methods are primarily evaluated on commonsense reasoning benchmarks, which we find to be unstable and weak indicators of compression quality.(see Appendix I) We instead evaluate on more challenging general-purpose benchmarks spanning mathematical reasoning, code generation, knowledge, and multilingual understanding following the evaluation settings in official technical reports (Qwen Team, 2024; Yang et al., 2025). The picture changes dramatically. Under a fair controlled comparison using a unified compression He et al. (2025b) framework, we evaluate activation-, score-, and magnitude-based methods at a fixed MoE compression ratio $p = 50\%$ (the fraction of routed-expert FFN parameters removed; defined in §3.1). As shown in Figure 2, all existing expert-level approaches suffer catastrophic performance collapse on benchmarks such as GSM8K, HumanEval, MBPP, and MATH.

We trace the failure of existing MoE compression methods to two factors. First, inaccurate importance metrics fail to estimate the importance of the parameters. Second, compression at an overly coarse granularity assumes capability is localized at the expert level, whereas in reality it is distributed across experts but concentrated in a small subset of intermediate dimensions. Our contribution is not the Fisher information itself but rather the choice of attributable intermediate dimension unit in an MoE, and aggregates parameter-level Fisher scores into intermediate dimension scores of experts. We use

[*]Equal Contribution



*Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 38550–38582
October 24-29, 2026 ©2026 Association for Computational Linguistics

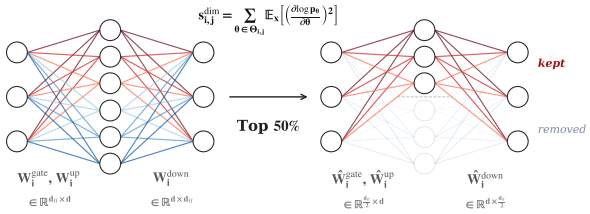


Figure 1: Intermediate dimension compression of a single MoE expert FFN. Edge colors encode the Fisher importance score $s_{i,j}^{\text{dim}}$ of each intermediate dimension (red = high, blue = low). The bottom $50\%$ of dimensions by Fisher score are removed (*faded*), reducing $W_i^{\text{gate}}, W_i^{\text{up}} \in \mathbb{R}^{d_{\text{ff}} \times d}$ and $W_i^{\text{down}} \in \mathbb{R}^{d \times d_{\text{ff}}}$ to $\hat{d}_{\text{ff}} = d_{\text{ff}}/2$ without discarding any expert.

Fisher importance as the underlying scoring tool to localize this attribution (§2) and to design a finer-grained compression method at the intermediate dimension level (§3).

**(1) Fisher importance as a unit attribution tool.** Prior methods rely on activation ratios, weight magnitude, or router scores, which we show are poor proxies for parameter importance. In contrast, empirical Fisher information performs substantially better as both a compression metric and an attribution tool. We validate this through three lines of evidence: (a) Fisher importance outperforms existing metrics under controlled comparison, (b) zeroing just 12 out of 1.35 million intermediate dimensions identified by Fisher importance destroys mathematical reasoning while preserving general knowledge, and (c) removing the bottom 50% of dimensions preserves overall performance (§2).

**(2) Fine-grained intermediate dimension compression.** Prior methods operate at the level of entire experts. However, capability in MoE models is distributed across experts but concentrated in a small subset of intermediate dimensions. Removing entire experts therefore discards critical dimensions alongside redundant ones. We close this expert-vs-intermediate-dimension localization gap by proposing **Fisher-MoE**, which converts the per-dimension attribution above into a structurally smaller MoE: it performs fine-grained compression within each expert by physically resizing the rows of $W^{\text{gate}}, W^{\text{up}}$ and the columns of $W^{\text{down}}$ corresponding to low-Fisher-score dimensions (§3).

Our key contributions are:

- We report a structural property of MoE models in which capability is not localized at the expert level but instead concentrated in a small subset of intermediate dimensions distributed across experts. This suggests one reason for the failure of expert-level MoE compression methods.
- We define the intermediate dimension as the attributable structure and use Fisher information as a tool for characterizing this structure. We empirically show that Fisher importance identifies critical and redundant intermediate dimension.
- We propose **Fisher-MoE**, a fine-grained compression method that operates at the intermediate-dimension level instead of expert-level. At a $p = 50\%$ compression ratio, it preserves downstream performance while improving inference throughput by $21\%$.

## 2 Model Capability Attribution

Can we find an importance metric that ranks parameters by their impact on model capability? In this section, we suggest the *empirical Fisher information* as a better metric. We first derive Fisher importance from this expansion and contrast it with prior heuristics (§2.1) then provide three lines of empirical evidence that this score is a useful ranking signal: it outperforms all alternatives (§2.1), masking few Fisher ranked critical dimensions collapses generation-heavy tasks while sparing knowledge tasks (§2.2), and the set of Fisher-important dimensions shared across tasks is small and removing it collapses every task (§2.3). For brevity, we group benchmarks into four categories: Knowledge covers `MMLU`, `CEval` and `CMMLU`; Code covers `HumanEval` and `MBPP`; Reasoning is `BBH`; Math covers `MATH` and `GSM8K`.

### 2.1 Empirical Fisher Information

Let $p_\theta(y \mid x)$ be the model's output distribution and $\mathcal{D}$ a calibration dataset of $N$ samples with sequence length $T$. We denote by $\mathcal{L}(x, y) = -\log p_\theta(y \mid x)$ the loss. Each expert $E_i$ is a gated FFN

$$E_i(\mathbf{x}) = W_i^{\text{down}} \left(\sigma(W_i^{\text{gate}}\mathbf{x}) \odot W_i^{\text{up}}\mathbf{x}\right), \quad (1)$$

with $W_i^{\text{gate}}, W_i^{\text{up}} \in \mathbb{R}^{d_{\text{ff}} \times d}$, $W_i^{\text{down}} \in \mathbb{R}^{d \times d_{\text{ff}}}$, and $\sigma(\cdot)$ an element-wise nonlinearity (e.g., SiLU).

**Empirical Fisher approximation.** A natural way to ask "how important is parameter $\theta_i$?" is to ask how much the predictive distribution moves when we perturb it. For a small $\delta \in \mathbb{R}^{|\theta|}$, a second-order Taylor expansion of the KL divergence between the unperturbed and perturbed models yields

$$\mathbb{E}_x \mathrm{KL}[p_\theta \,\|\, p_{\theta+\delta}] = \delta^\top F_\theta \, \delta + O(\|\delta\|^3), \quad (2)$$

where $F_\theta \in \mathbb{R}^{|\theta|\times|\theta|}$ is the *Fisher Information Matrix*, Letting $g_\theta(x, y) := \nabla_\theta \log p_\theta(y \mid x)$ denote the per-token score function,

$$F_\theta = \mathbb{E}_{x,y}\Big[g_\theta(x, y)\, g_\theta(x, y)^\top\Big]. \quad (3)$$

The full $F_\theta$ is intractable for billion-parameter models. We adopt the *empirical Fisher*: we replace $\mathbb{E}_{y\sim p_\theta(y|x)}$ by the ground-truths in $\mathcal{D}$, yielding

$$\hat{F}_\theta = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} (\nabla_\theta \log p_\theta(y \mid x))^2 \quad (4)$$

Throughout the rest of the paper we refer to this score simply as the *Fisher*. When we apply this metric to MoE compression we call the resulting method Fisher-MoE.

**Baselines.** Prior MoE compression instead uses one of three metrics:

- *Activation ratio:* $s_i^{\text{act}} = \frac{1}{NT}\sum_{(\mathbf{x},t)\in\mathcal{D}} \mathbb{1}[i \in T(\mathbf{x}_t)]$, the fraction of tokens routed to expert $i$ (Muzio et al., 2024; Lu et al., 2024a).
- *Router score:* $s_i^{\text{score}} = \frac{1}{NT}\sum g_i(\mathbf{x}_t)$, the average gating weight (Xie et al., 2024).
- *Magnitude:* $s_i^{\text{mag}} = \|W_i^{\text{gate}}\|_F + \|W_i^{\text{up}}\|_F + \|W_i^{\text{down}}\|_F$, a data-independent weight norm (Lee et al., 2025; Yang et al., 2024b).

**Extension to intermediate dimensions.** A key practical advantage of Fisher importance is that the same derivation applies whether $W$ is a router weight matrix, a full expert FFN, or a single intermediate dimension. We instantiate Eq. 4 at three granularities used throughout the paper: ***(1) Router-level Fisher***: $W$ is the row of the router gate corresponding to expert $i$; ***(2) Expert-level Fisher***: $W$ is the union of $W_i^{\text{gate}}, W_i^{\text{up}}, W_i^{\text{down}}$ for expert $i$; ***(3) Intermediate dimension Fisher***: $W$ is the rows of $W_i^{\text{gate}}, W_i^{\text{up}}$ and the column of $W_i^{\text{down}}$ corresponding to a single intermediate dimension $j$ of expert $i$. $G_i^m := \nabla_{W_i^m}\mathcal{L}$ for $m \in \{g, u, d\}$,

$$s_{i,j}^{\text{Fisher}} = \frac{1}{d}\sum_k \left[(G_i^g)^2_{j,k} + (G_i^u)^2_{j,k} + (G_i^d)^2_{k,j}\right] \quad (5)$$

averaged over calibration samples; the sum runs over the hidden dimension $k \in [d]$ and the denominator $d$ normalizes by the total number of parameters tied to dimension $j$. The intermediate dimension form is what enables both fine-grained compression (removing low-scoring dimensions, §3) and fine-grained attribution (identifying high-scoring critical dimensions, §2.2).

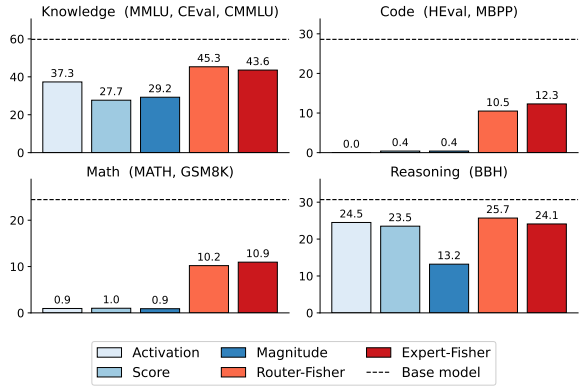


Figure 2: Expert-level pruning with existing important metrics on Qwen1.5-MoE at 50% compression ratio.

We compare all four importance metrics under expert-level compression at MoE compression ratio $p = 50\%$. To ensure a fair comparison, we use the unified compression framework of (He et al., 2025b): all methods share identical calibration data (GSM8K training set, 128 samples), compression procedure, and evaluation protocol on Qwen1.5-MoE; only the importance metric differs. Fisher importance dominates across all tasks (Figure 2). Full results can be found in Table 10.

## 2.2 Can Fisher Importance Identify Task-Critical Parameters?

| Setting | Knowledge | Code | Reasoning | Math | |
|---|---|---|---|---|---|
| | | | | MATH | GSM8K |
| **Base model** | 59.8 | 28.6 | 30.7 | 13.0 | 35.9 |
| *Top-masking* | | | | | |
| **Mask-Top-12** | 56.7 | 10.7 | 23.6 | 1.1 | 0.8 |
| *Critical vs Redundant Dimension Removal* | | | | | |
| **Remove redundant** ($\mathcal{D}_\cap$) | 61.1 | 28.5 | 31.2 | 8.7 | 31.2 |
| **Remove critical** ($\mathcal{K}_\cap$) | 1.7 | 0.0 | 0.2 | 0.0 | 0.0 |

Table 1: Impact of masking top-12 and removing critical/redundant intermediate dimensions on performance.

We rank all ∼1.35M MoE FFN intermediate dimensions by Fisher score (computed on 128 `GSM8K` training samples) and zero-mask the top fraction. Masking just 12 of 1.35M intermediate dimensions (0.001%) collapses `GSM8K` from 35.9% to 0.8% and `MATH` from 13.0% to 1.1%, with code and `BBH` also degrading sharply, while multiple-choice knowledge tasks (`MMLU/CEval/CMMLU`) largely retain base (Table 1; full breakdown in Table 11).

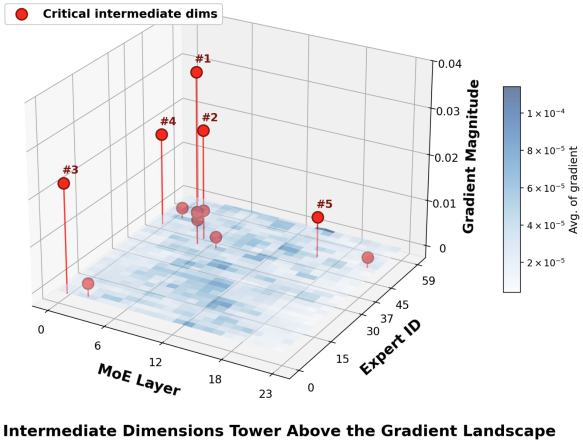


Figure 3: Fisher scores of intermediate dimensions across experts and layers in Qwen1.5-MoE.

Appendix J confirms none of the residual 0.8% on GSM8K reflects real computation – all 10 "correct" answers are coincidental digit matches. Figure 3 shows the underlying cause: the Fisher distribution is extremely heavy-tailed, with these twelve dimensions $\sim$1000$\times$ above the population mean.

We further measure each dimension's mean forward activation magnitude. The population is sharply skewed: the top-12 Fisher dimensions carry mean 7–450$\times$ the median activation of all parameters. (Appendix B) They match the *massive-activation* signature of Sun et al. (2024a), further suggest Fisher identify critical parameters.

To understand why masking only twelve dimensions devastates math reasoning, we trace the failure into the model's attention dynamics. *Attention sink* (Xiao et al., 2024) stabilises autoregressive decoding. Masking the top-12 collapses mean <BOS> attention in the mid-stack (layer 7$\sim$17, where the sink dominates) by $\sim$70% reduction (Appendix C). This directly explains the dissociation in Table 1: multi-step generation depends on the sink-stabilised token-to-token dynamics and degenerates into the echo and garbled outputs of Appendix J, whereas MCQ tasks need only a single-token classification and are largely sink-insensitive.

The cross-domain analysis from §H suggests why removing GSM8K-calibrated critical dimensions selectively destroys math reasoning. Table 24 shows that the pairwise overlap of Fisher-important dimensions between tasks exhibits clear domain structure: math tasks (GSM8K/MATH) share 65.3% of their important dimensions. The high overlap explains why removing GSM8K-calibrated critical dimensions also destroys MATH performance, while the lower overlap with knowledge tasks ($\sim$55%) suggests why MMLU/CEval/CMMLU remain intact.

### 2.3 Identify Universally Indispensable Dimensions

The domain-overlap analysis identifies a set of intermediate dimensions retained by all eight evaluation tasks under $p = 50\%$ ($\mathcal{K}_\cap$, 4.88% of parameters). Removing the universally *kept* dimensions collapses every task to near-zero accuracy. Removing the universally *discarded* dimensions ($\mathcal{D}_\cap$, 4.01% of parameters) preserves near-full performance across all tasks (Table 1), confirming these dimensions are genuinely redundant. This result, combined with §2.2, leads to a key insight for compression: removing the long redundant tail of intermediate dimensions ranked by Fisher preserves model performance. We leverage this in §3.

## 3 Keep the Experts, Slim Their FFNs

Section 2 showed that Fisher ranks parameters in a way that aligns with downstream impact: removing the top-ranked most disrupts generation, while the bottom-ranked form a long redundant tail. In this section, we leverage this ranking for compression. We first formalize prior expert-level compression and our finer-grained intermediate dimension alternative (§3.1), then demonstrate that intermediate dimension compression substantially outperforms expert-level methods (§3.2), and finally quantify the deployment benefits (§3.3).

### 3.1 From Expert-Level to Intermediate Dimension Compression

A Mixture-of-Experts layer consists of $n$ expert combined through a learned routing mechanism. Given an input $\mathbf{x} \in \mathbb{R}^d$, the MoE layer output is

$$f(\mathbf{x}; \Phi) = \sum_{i \in T(\mathbf{x})} g_i(\mathbf{x})\, E_i(\mathbf{x}; \theta_i), \qquad (6)$$

where $T(\mathbf{x})$ denotes the set of top-$k$ experts selected by the router, $g_i(\mathbf{x})$ is the routing weight for expert $i$, and $\Phi = \{(\theta_i, W_i, b_i)\}_{i=1}^n$ collects all layer parameters. Let $\mathcal{P}_{\text{MoE}}$ be all of the experts FFN weights, and $\hat{\mathcal{P}}_{\text{MoE}}$ what remains after compression. We define the ***MoE compression ratio*** as $p = 1 - \frac{|\hat{\mathcal{P}}_{\text{MoE}}|}{|\mathcal{P}_{\text{MoE}}|}$. At $p = 50\%$, expert-level methods remove half of the experts, and intermediate dimension compression reduces $d_{\text{ff}}$ inside experts. Because the MLP weights dominate LLM parameter counts, while attention, embeddings, router, and

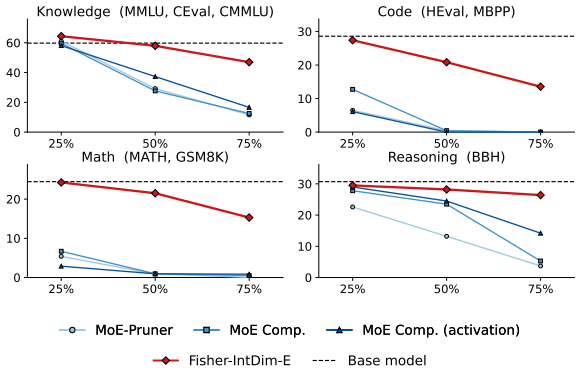


Figure 4: Performance of Qwen1.5-MoE under expert-level compression at three drop ratios (25%, 50%, 75%).

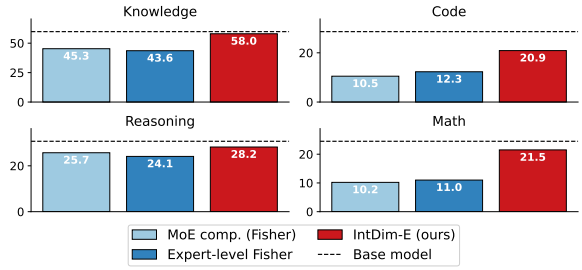


Figure 5: Expert-level vs. intermediate dimension compression at $p = 50\%$.

layer norms are never compressed, $p = 50\%$ still translates to a 43–48% whole-model compression rate $p_{\text{model}}$ across backbones (reported in Table 37).

**Expert-level compression.** Prior methods select a subset $\mathcal{S} \subset \{1, \ldots, n\}$ of experts to remove in each MoE layer. The compressed layer becomes

$$\hat{f}(\mathbf{x}) = \sum_{i \in T(\mathbf{x}) \setminus \mathcal{S}} \hat{g}_i(\mathbf{x})\, E_i(\mathbf{x}; \theta_i), \tag{7}$$

where $\hat{g}_i$ denotes re-normalized routing weights. Expert selection uses activation ratio, router score, weight magnitude, or Fisher.

**Proposed Intermediate dimension compression.** Instead of removing entire experts, we operate at the granularity of individual FFN intermediate dimensions. For each expert $E_i$ in each MoE layer, we use the Fisher importance score $s^{\text{dim}}_{i,j}$ (Eq. 5, formulated in §2.1) to rank every intermediate dimension $j \in \{1, \ldots, d_{\text{ff}}\}$ and remove the lowest-scoring fraction $p$. The compressed expert becomes

$$\hat{E}_i(\mathbf{x}) = \hat{W}^{\text{down}}_i \left( \sigma(\hat{W}^{\text{gate}}_i \mathbf{x}) \odot \hat{W}^{\text{up}}_i \mathbf{x} \right), \tag{8}$$

where $\hat{W}^{\text{gate}}_i, \hat{W}^{\text{up}}_i \in \mathbb{R}^{\hat{d}_{\text{ff}} \times d}$ and $\hat{W}^{\text{down}}_i \in \mathbb{R}^{d \times \hat{d}_{\text{ff}}}$ are obtained by physically removing the rows of $W^{\text{gate}}_i, W^{\text{up}}_i$ and the corresponding columns of $W^{\text{down}}_i$. The removal is determined by the Fisher $s^{\text{dim}}_{i,j}$ together with a budget rule that fixes *where* the budget $p$ is spent. We instantiate three rules: **IntDim-E** keeps the top $(1-p)$ fraction of dimensions within each expert; **IntDim-L** pools dimensions across experts in a layer and keeps the top $(1-p)$ fraction per layer; **IntDim-G** pools dimensions across the whole model and keeps the top $(1-p)$ globally. All three share the same score and budget to isolate the effect of allocation flexibility. They reduce the actual parameter count and computation per expert and preserve the model's routing behavior. We choose Fisher importance as our scoring criterion, as justified in §2.

### 3.2 Intermediate Dimension Compression Outperforms Expert-Level Methods

We compare expert-level and intermediate dimension compression at the same compression ratio $p = 50\%$ on Qwen1.5-MoE-A2.7B across eight general-purpose benchmarks. Results on additional backbones, including OLMoE, Qwen3-30B, and Qwen3.5-35B, are provided in §4. Both use Fisher and domain-specific calibration data (`GSM8K` 128 training samples). At $p = 50\%$, expert-level compression removes half routed experts per layer; IntDim-E reduces $d_{\text{ff}}$ by half within each expert.

Table 5 (Full results in Table 13) reveals that intermediate dimension compression dramatically outperforms expert-level compression including Expert-level Fisher across all benchmarks. We further scale our experiments across different compression ratios in Fig. 4 (Full results in Table 12): with $p = 25\%, 50\%, 75\%$, intermediate dimension removal consistently retains base model performance and outperform expert-level methods. For a finer-grained evaluation within the knowledge category (`MMLU`, `CEval`, and `CMMLU`), we additionally evaluate compression ratios of 40%, 60%, and 90%, with the full per-benchmark curves provided in Appendix Fig. 6. In particular, comparisons with Expert-level Fisher highlight the advantage of fine-grained compression.

The core reason for this advantage is that knowledge in MoE models is distributed across all experts in a long-tailed manner, rather than concentrated in a few (He et al., 2026). Each expert contains both essential and redundant intermediate dimensions. Expert-level compression must discard en-

tire experts including their essential components. Whereas intermediate dimension compression selectively removes only the least important computation within each expert, precisely the dimensions that §2.3 showed are genuinely dispensable.

### 3.3 Deployment Cost and Inference Speedup

| Metric | Base | Compressed | Change |
|---|---|---|---|
| Latency ↓ | 2.233 s | 1.839 s | 1.21× ↓ |
| Throughput ↑ | 7,336 tok/s | 8,909 tok/s | 1.21× ↑ |
| KV-cache budget ↑ | 187K | 251K | +33.7% |
| VRAM ↓ | 26.67 GiB | 15.09 GiB | −43.4% |
| Active params ↓ | 2.689 B | 2.274 B | −15.4% |

Table 2: Inference efficiency of 50% compressed Qwen1.5-MoE on H100.

Intermediate dimension compression achieves ***(1) parameter reduction*** compared to the base model: the gate, up, and down projection matrices are physically resized, where $\hat{W}_i^{\text{gate}}, \hat{W}_i^{\text{up}} \in \mathbb{R}^{\hat{d}_{\text{ff}} \times d}, \hat{W}_i^{\text{down}} \in \mathbb{R}^{d \times \hat{d}_{\text{ff}}}$ are compressed from $W_i^{\text{gate}}, W_i^{\text{up}} \in \mathbb{R}^{d_{\text{ff}} \times d}, W_i^{\text{down}} \in \mathbb{R}^{d \times d_{\text{ff}}}$, producing a smaller model. ***(2) Inference speedup*** compared to the base model and other expert-level removal baselines: Expert removal methods (Muzio et al., 2024; Lu et al., 2024a; Chen et al., 2022; Li et al., 2023; Chen et al., 2025) discard entire experts from MoE layers but keep the number of activation parameters per token the same. The FFN intermediate dimension reduction translates to reduced active-parameter savings per token instead.

We benchmark the 50% compress ratio Qwen1.5-MoE against the base model on a single NVIDIA H100 GPU using vLLM with bf16 precision.[1] Concretely, each layer goes from 86.1M to 68.8M parameters (a 20% per-layer reduction). The theoretical decode-time speedup for Qwen1.5-MoE is $2.689\text{B}/2.274\text{B} \approx 1.18\times$. Table 2 shows that the IntDim-E compressed model achieves a $1.21\times$ wall-clock speedup, slightly exceeding the theoretical estimate. The additional gain comes from the freed VRAM being automatically converted by vLLM into a larger KV cache, increasing the request capacity by 34%. The VRAM and disk footprint reduces from 27 GB to 16 GB.

[1]We greedy decode 1024 prompts and *max_new_tokens* = 256. We use 1024-prompt warm-up precedes each timed run and `gpu_memory_utilization=0.85`.

## 4 Experiments

In Subsection §4.1, we study how different MoE compression strategies behave when directly applied to the base model without any post-training. We compress Qwen1.5-MoE-A2.7B and OLMoE-1B-7B-0125 at $p = 50\%$ under task-matched domain calibration, and report performance across the eight general-task benchmarks above to isolate the effect of the importance metric and compression granularity. In Subsection §4.2, we study whether the granularity advantage transfers to long-CoT mathematical reasoning at frontier scale on Qwen3-30B-A3B and Qwen3.5-35B-A3B with 128 `Stanford-S1` calibration data. In Subsection §4.3, we study whether post-trained compression models can achieve competitive results compared to full post-trained models. In Subsection §4.4, we study the out-of-domain generalization of single-domain calibration. We use `GSM8K` calibration for all methods and evaluate the resulting model on the seven non-math benchmarks to test whether the Fisher signal preserves general capability when the calibration distribution is narrow. In Subsection §4.5, we empirically show that Fisher-MoE composes with quantization to further reduce model deployment cost. We apply 4-bit AWQ (Lin et al., 2024) on top of Fisher-MoE at $p = 50\%$ and compare the accuracy and footprint of *Ours+AWQ* against *Base+AWQ* to verify that the two reductions compose without amplifying quantization sensitivity.

We distinguish three calibration regimes here: same-domain and same-format, same-domain and different-format, and different-domain and different-format. The first matches the target deployment setting, while the latter two test transfer across format or domain shifts.

After evaluating different compression ratios in Fig. 4 ($p = 25\%, 50\%, 75\%$), we fix the MoE compression ratio to $p = 50\%$ (defined in §3.1, the *whole-model* parameter reduction is ∼45% across four backbones; see Table 37) for the remainder of this section, and focus on scaling across models and benchmarks, along with generalization and compatibility. Detailed experimental settings, including models and dataset, baselines, calibration data, computational resources, and training and evaluation framework are in Appendix E. We include an ablation study and cost analysis on the calibration sample size $N \in 32, 64, 128, 256, 512$ in Appendix F, and use $N = 128$; all methods use

Table 3: Performance of different pruning strategies at MoE compression ratio $p = 50\%$. **Upper:** Qwen1.5-MoE-A2.7B and OLMoE-1B-7B on general benchmarks. **Lower:** Qwen3-30B-A3B and Qwen3.5-35B-A3B on frontier long-CoT math reasoning. Calibration uses the corresponding domain training data. Here, "HE" is short for HumanEval, "MA" is short for MultiArith, and "Reas." is short for Reasoning.

| Method | *Qwen1.5-MoE-A2.7B* | | | | | | | | | | *OLMoE-1B-7B* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Knowledge | | | Coding | | Math | | | Reas. | Avg. | Knowledge | | | Coding | | Math | | | Reas. | Avg. |
| | MMLU | CEval | CMMLU | HE | MBPP | MATH | GSM8K | MR | BBH | | MMLU | CEval | CMMLU | HE | MBPP | MATH | GSM8K | MA | BBH | |
| Base Model | 59.3 | 59.3 | 60.7 | 32.6 | 24.6 | 13.0 | 35.9 | 76.7 | 30.7 | 43.6 | 53.5 | 33.1 | 33.0 | 9.8 | 15.2 | 11.6 | 53.2 | 87.5 | 26.5 | 35.9 |
| MoE comp. (activation) | 26.3 | 42.5 | 43.1 | 0.0 | 0.0 | 0.0 | 1.9 | 3.3 | 24.5 | 15.7 | 33.2 | 22.5 | 26.0 | 9.2 | 11.9 | 1.5 | 5.0 | 22.2 | 22.6 | 17.1 |
| MoE comp. | 37.2 | 13.5 | 32.3 | 0.0 | 0.8 | 0.0 | 1.9 | 2.0 | 23.5 | 12.4 | 31.6 | 22.6 | 27.1 | 9.2 | 10.6 | 3.5 | 7.1 | 17.3 | 24.7 | 17.1 |
| MoE-Pruner[2] | 18.6 | 33.3 | 35.8 | 0.8 | 0.0 | 0.1 | 1.7 | 2.7 | 13.2 | 11.8 | 17.3 | 12.9 | 19.0 | 0.0 | 0.0 | 0.0 | 2.1 | 2.7 | 12.2 | 7.4 |
| MoE comp. (Fisher) | 39.4 | 48.8 | 47.7 | 12.9 | 8.1 | 2.0 | 16.9 | 15.5 | 25.7 | 24.1 | 33.8 | 27.0 | 29.0 | 5.5 | 9.1 | 1.9 | 5.2 | 26.5 | 24.0 | 18.0 |
| Expert-level Fisher | 37.9 | 48.6 | 44.2 | 12.9 | 11.7 | 2.4 | 18.0 | 37.7 | 24.1 | 26.4 | 34.4 | 27.7 | 27.6 | 7.3 | 10.1 | 1.7 | 6.1 | 21.5 | 25.4 | 18.0 |
| **Fisher-IntDim-E (ours)** | 50.3 | 62.0 | 61.6 | 21.2 | 20.5 | 8.0 | 35.0 | 61.5 | 28.2 | 38.7 | 39.9 | 29.2 | 29.9 | 9.8 | 7.3 | 2.3 | 24.9 | 75.7 | 27.3 | 27.4 |
| **Fisher-IntDim-L (ours)** | 49.1 | 57.5 | 62.0 | 30.5 | 22.8 | 12.6 | 22.0 | 89.8 | 28.3 | 41.6 | 40.4 | 30.8 | 31.4 | 9.8 | 17.0 | 7.8 | 44.3 | 87.7 | 25.1 | 32.7 |
| **Fisher-IntDim-G (ours)** | 49.0 | 61.3 | 64.9 | 34.2 | 24.1 | 9.8 | 27.2 | 92.2 | 28.2 | 43.4 | 40.4 | 31.4 | 31.7 | 10.4 | 15.7 | 6.9 | 41.9 | 81.0 | 25.2 | 31.6 |

| Method | *Qwen3-30B-A3B* | | | | | | | *Qwen3.5-35B-A3B* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AIME25 | AIME26 | Olympiad | MATH-500 | GSM8K | GPQA-D | Avg. | AIME25 | AIME26 | Olympiad | MATH-500 | GSM8K | GPQA-D | Avg. |
| Base Model | 33.3 | 50.0 | 49.6 | 64.2 | 61.9 | 48.5 | 51.3 | 66.7 | 66.7 | 77.9 | 92.4 | 48.8 | 72.2 | 70.8 |
| MoE-Pruner | 0.0 | 0.0 | 0.3 | 2.4 | 0.8 | 5.1 | 1.4 | 13.3 | 10.0 | 22.1 | 34.8 | 17.4 | 19.7 | 19.6 |
| MoE comp. | 0.0 | 0.0 | 0.9 | 1.8 | 2.3 | 10.6 | 2.6 | 0.0 | 0.0 | 0.0 | 0.2 | 2.1 | 12.6 | 2.5 |
| MoE comp. (activation) | 0.0 | 0.0 | 1.2 | 1.6 | 2.8 | 2.5 | 1.4 | 0.0 | 0.0 | 0.2 | 0.2 | 2.3 | 9.6 | 2.0 |
| MoE comp. (Fisher) | 46.7 | 50.0 | 55.0 | 76.4 | 92.3 | 38.9 | 59.9 | 80.0 | 76.7 | 79.8 | 91.8 | 42.2 | 71.7 | 73.7 |
| Expert-level Fisher | 36.7 | 53.3 | 54.9 | 76.2 | 93.4 | 49.0 | 60.6 | 80.0 | 76.7 | 81.9 | 92.4 | 50.7 | 69.7 | 75.2 |
| **Fisher-IntDim-G (ours)** | 60.0 | 56.7 | 58.6 | 74.6 | 81.4 | 50.0 | 63.5 | 76.7 | 73.3 | 79.5 | 92.4 | 62.6 | 70.2 | 75.8 |

the same calibration data. 128 samples cost less than 30 seconds for Fisher calculation on an H100 node for the 30B models. Further comparisons with dense pruning and sparsity baselines are in Appendix L, though they fall outside the scope of MoE architectures. To fully evaluate the performance, we further report the per-benchmark retention, safety, instruction-following, and perplexity in Appendix D.

### 4.1 Compression on Base Models: General Tasks

We compress each base model at the MoE compression ratio $p = 50\%$ and evaluate across the eight general-purpose benchmarks. This isolates the effect of the compression algorithm: any performance loss reflects information that the importance metric and granularity failed to preserve.

Table 3 shows a consistent picture across both backbones. Expert-level baselines, regardless of importance metric, suffer catastrophic collapse on generation-heavy tasks, including `GSM8K`, `MATH`, and `HumanEval`. Fisher-MoE, on the other hand, preserves generation capability: it largely retains capability in math and code reasoning datasets. The advantage holds across knowledge tasks too.

To disentangle importance scoring from pruning granularity, we add a matched factorial at $p = 50\%$ (full results in Appendix Table 14). Granularity helps under every metric, Fisher helps at both granularities, and their combination achieves the best performance, showing that neither factor alone explains the gains.

Additionally, enabling more flexible allocation of intermediate dimension leads to further gains. Specifically, Fisher-IntDim-G outperforms Fisher-IntDim-E on both OLMoE and Qwen1.5-MoE. In the remainder of this section, we adopt Fisher-IntDim-G for scaling experiments, including larger models, long chain-of-thought (CoT) math reasoning, and Supervised Fine-tuning (SFT).

On `CEval`, `CMMLU`, and `MultiArith`, Fisher-MoE matches or modestly exceeds the uncompressed base model. We interpret these gains as evidence that Fisher-based pruning can sometimes remove dimensions associated with brittle shortcut behavior or out-of-distribution output formats, thereby revealing capabilities already present in the base model rather than introducing new ones. A similar pattern is observed for the Qwen3 and Qwen3.5 models in §4.2. We further analyze this phenomenon quantitatively in Appendix K. Compression reduces generation-time biases that suppress useful computation already available in the original model. Finally, we show that removing entire experts or attention heads causes severe degradation on generation-heavy benchmarks, while FFN intermediate dimension compression is substantially more robust (Appendix Q).

*(a) Qwen1.5-MoE-A2.7B*

| Method | MATH | GSM8K | MultiArith |
|---|---|---|---|
| Base + SFT | 15.9 | 50.8 | 87.8 |
| MoE-Pruner + SFT | 3.7 | 40.6 | 35.8 |
| MoE comp. + SFT | 4.9 | 30.3 | 44.7 |
| MoE comp. (activation) + SFT | 4.8 | 28.5 | 47.7 |
| MoE comp. (Fisher) + SFT | 14.3 | 46.0 | 81.8 |
| Expert-level Fisher + SFT | 15.4 | 46.4 | 84.8 |
| **Fisher-IntDim-G + SFT (ours)** | 17.6 | 52.0 | 88.0 |

*(b) OLMoE-1B-7B-0125*

| Method | MATH | GSM8K | MultiArith |
|---|---|---|---|
| Base + SFT | 12.4 | 45.1 | 92.5 |
| MoE-Pruner + SFT | 2.3 | 15.2 | 39.5 |
| MoE comp. + SFT | 9.4 | 30.6 | 87.8 |
| MoE comp. (activation) + SFT | 9.8 | 31.8 | 87.0 |
| MoE comp. (Fisher) + SFT | 11.4 | 37.1 | 89.3 |
| Expert-level Fisher + SFT | 10.5 | 39.1 | 89.2 |
| **Fisher-IntDim-G + SFT (ours)** | 11.9 | 43.8 | 92.3 |

Table 4: Compression at MoE ratio $p = 50\%$ followed by SFT on domain training data. *Base + SFT* is the uncompressed reference ceiling.

## 4.2 Compression on Base Models: Math Reasoning at Larger Scale

We next test whether the granularity advantage carries over to long-CoT math reasoning at larger scale, where errors compound across many decoding steps and any loss of generation capability is unforgiving. We compress Qwen3-30B-A3B and Qwen3.5-35B-A3B at the same MoE compression ratio $p = 50\%$ and evaluate on five long-CoT reasoning benchmarks in Tab. 3.

## 4.3 Fine-Tuning Enabled Domain Enhancement

The base-model results above measure the naive case: compressed weights deployed without any further training. In practice, a compressed model is often used as the starting point for domain SFT. The relevant question for deployment is therefore: *does compressing first then SFTing recover the base + SFT ceiling?* We compress at $p = 50\%$, then SFT on the corresponding domain training data, and compare against the uncompressed base + SFT.

Results in Tab. 4 show that using half the FFN parameters, after SFT, Fisher-MoE reaches competitive results compared with the uncompressed Base + SFT ceiling on `GSM8K`, `MultiArith`, and `MATH`. In contrast, the activation-, score-, and magnitude-based compression methods significantly underperform even after SFT.

## 4.4 Out-of-Domain Generalization

A reasonable concern with Fisher importance is that calibration on one domain may bias the retained dimensions toward that domain, hollowing out general capability. We test this directly: we calibrate on math (`GSM8K`, 128 samples), apply 50% intermediate dimension compression, and evaluate the resulting model on seven Out-of-Domain (OOD) task categories. Then we further compare the OOD performance of 50% intermediate dimension compression with other baselines. Results in Tab. 5 show that for both in-domain and OOD tasks, Fisher-IntDim-E dominates the activation-, score-, and magnitude-based baselines by large margins.

[2]For OLMoE-1B-7B, MoE-Pruner uses low-magnitude weights as the importance metric, since the original high-magnitude variant collapses accuracy across all tasks.

Table 5: Evaluation of Qwen1.5-MoE compressed models. `GSM8K` is used for calibration. In-domain tasks include arithmetic reasoning (`MATH/GSM8K/MultiArith`); out-of-domain results are grouped into Knowledge (`MMLU/CEval/CMMLU`), Code (`HumanEval/MBPP`) and Reasoning (`BBH`). Full results in Table 15.

| Method | In-domain (Math) | | | Out-of-domain | | |
|---|---|---|---|---|---|---|
| Qwen1.5-MoE-A2.7B | GSM8K (Calib) | MATH | MultiArith | Knowledge | Code | Reasoning |
| MoE-Pruner | 1.7 | 0.0 | 1.8 | 29.2 | 0.3 | 13.2 |
| MoE comp. | 1.9 | 0.3 | 0.3 | 37.0 | 0.6 | 22.1 |
| MoE comp. (activation) | 1.9 | 0.2 | 0.8 | 38.1 | 0.2 | 12.3 |
| MoE comp. (Fisher) | 16.9 | 0.7 | 26.2 | 37.0 | 4.2 | 23.0 |
| Expert-level Fisher | 18.0 | 0.7 | 31.7 | 38.7 | **5.8** | **24.6** |
| **Fisher-IntDim-E (ours)** | **35.0** | **4.7** | **70.8** | **49.4** | **5.6** | **24.2** |

## 4.5 Compatibility with Quantization

Fisher-MoE reduces parameter count structurally; AWQ (Lin et al., 2024) reduces bit-width per parameter. The two operate on orthogonal axes of model footprint, so combining them should multiply the savings if neither destroys the capability the other depends on. We test this by applying AWQ (settings in Appendix §G) on top of IntDim-E at $p = 50\%$ and against AWQ on the base model.

Table 6: Stacking AWQ on top of Fisher-MoE at MoE ratio $p = 50\%$. Full results in Table 16.

| Method | Disk | VRAM | Knowledge | Code | Reasoning | Math | Avg. |
|---|---|---|---|---|---|---|---|
| Base Model | 26.7 | 58.1 | 59.8 | 28.6 | 30.7 | 41.9 | 40.3 |
| Base Model + AWQ | 7.9 | 8.6 | 50.2 | 12.8 | 22.0 | 11.6 | 24.2 |
| IntDim-E | 15.1 | 30.1 | 58.0 | 20.9 | 28.2 | 34.8 | 35.5 |
| **IntDim-E + AWQ** | 5.0 | 6.5 | 50.2 | 13.3 | 24.6 | 13.5 | 25.4 |

The two reductions compose: combining *IntDim-E* at $p = 50\%$ with 4-bit AWQ shrinks the deployed footprint of Qwen1.5-MoE-A2.7B from 26.67 GiB (bf16, base) to roughly 1/8 of that, while preserving

the capability profile reported in §3.3. The accuracy loss from *IntDim-E+AWQ* relative to *IntDim-E* matches the loss of *Base Model+AWQ* relative to *Base Model*, confirming that Fisher-MoE is compatible with quantization.

## 5 Conclusion

We revisit MoE compression through the lens of two questions: *which signal* should drive parameter selection, and *at what granularity* should compression operate. For the first, we show that the empirical Fisher information outperforms activation-, router-, and magnitude-based heuristics across backbone and benchmark (§2, §4.1. §4.2). For the second, we show that pushing compression below the expert level to FFN intermediate dimensions changes the regime (§3, §4.1). Beyond speedup and memory efficiency advantages (§3.3, §4.5), our Fisher-importance attribution analysis localizes a striking dissociation (§2.2), suggesting that intermediate dimension Fisher is a useful tool for both compression (§ 3) and understanding where capabilities live inside models (§ 2.2 2.3).

## Limitations

Due to limited computational resources, we did not extend the exploration of Fisher-MoE to 100 Billion+ parameters MoE backbones, nor did we run a full recovery study with longer post-training on larger and more diverse corpora to characterize how much of the residual gap to the uncompressed base model can be closed. We leave both directions as natural follow-ups for future work.

## Acknowledgments

We extend our gratitude to Yu (Ivy) Yang for her helpful suggestions on manuscript preparation and research presentation.

This research was conducted using the ORCHARD high-performance computing cluster operated by the Office of Research Computing and Data at Carnegie Mellon University. We gratefully acknowledge Carnegie Mellon University for providing access to these shared computing resources.

This research also used the Bridges-2 system at the Pittsburgh Supercomputing Center and the Delta advanced computing system. These resources are supported by the National Science Foundation under Grants No. 2138259, 2138286, 2138307, 2137603, and 2138296.

## References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. 2025. Retraining-free merging of sparse moe via hierarchical clustering. In *International Conference on Machine Learning*, pages 8594–8620. PMLR.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*.

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Hao Gu, Wei Li, Lujun Li, Qiyuan Zhu, Mark Lee, Shengjie Sun, Wei Xue, and Yike Guo. 2025. Delta decompression for moe-based llms compression. *arXiv preprint arXiv:2502.17298*.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

B. Hassibi, D.G. Stork, and G.J. Wolff. 1993. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1.

Babak Hassibi and David Stork. 1992. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5.

Haoze He, Xingyuan Ding, Xuan Jiang, Xinkai Zou, Alex Cheng, Yibo Zhao, Juncheng Billy Li, and Heather Miller. 2026. Preserving long-tailed expert information in mixture-of-experts tuning. *arXiv preprint arXiv:2604.23036*.

Haoze He, Juncheng Billy Li, Xuan Jiang, and Heather Miller. 2025a. Sparse matrix in large language model fine-tuning. *Preprint*, arXiv:2405.15525.

Shwai He, Daize Dong, Liang Ding, and Ang Li. 2025b. Towards efficient mixture of experts: A holistic study of compression techniques. *Transactions on Machine Learning Research*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. 2019. Limitations of the empirical fisher approximation for natural gradient descent. *Advances in neural information processing systems*, 32.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.

Jaeseong Lee, Seung-won Hwang, Aurick Qiao, Daniel F Campos, Zhewei Yao, and Yuxiong He. 2025. STUN: Structured-then-unstructured pruning for scalable MoE pruning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13660–13676, Vienna, Austria. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *arXiv preprint arXiv:2310.01334*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024a. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6172, Bangkok, Thailand. Association for Computational Linguistics.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024b. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6172.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*.

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, and 5 others. 2024. Olmoe: Open mixture-of-experts language models. *Preprint*, arXiv:2409.02060.

Alexandre Muzio, Alex Sun, and Churan He. 2024. Seer-moe: Sparse expert efficiency through regularization for mixture-of-experts. *arXiv preprint arXiv:2404.05089*.

Qwen Team. 2024. Qwen1.5-MoE: Matching 7b model performance with 1/3 activated parameters. *Qwen Blog*.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. {Zero-offload}: Democratizing {billion-scale} model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1743–1752.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024a. Massive activations in large language models. In *Conference on Language Modeling (COLM)*.

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

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. 2024. Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. 2024b. MoE-i$^2$: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10456–10466, Miami, Florida, USA. Association for Computational Linguistics.

## A Related Work

Prior MoE compression methods can be grouped by their *importance metric* (how compression targets are selected, we discussed in § 1).

**Importance metrics.** Across these strategies, methods differ in how they select which experts or parameters to compress. **Activation-ratio-based** methods measure routing frequency (Muzio et al., 2024; Lu et al., 2024a; Chen et al., 2022). **Router-score-based** methods use average gating weights (Xie et al., 2024; Gu et al., 2025). **Magnitude-based** methods rank by weight norms (Lee et al., 2025; Yang et al., 2024b). MoE-Pruner (Xie et al., 2024) combines gated values with weight magnitudes. Our work demonstrates that **gradient-based** importance—which directly measures each parameter's contribution to the loss—substantially outperforms all of these alternatives (§2.1). Moreover, all prior MoE compression methods operate at the granularity of entire experts or expert blocks; our intermediate dimension compression (§3) represents a strictly finer granularity that better preserves the distributed knowledge structure of MoE models.

**Model Attribution** Our Fisher-importance-based model attribution analysis relates to the broader literature on understanding which model components are responsible for specific capabilities. The discussion about parameter importance can trace back to Optimal Brain Damage (LeCun et al., 1989) and Optimal Brain Surgeon (Hassibi et al., 1993), which use second-order information (Hessian diagonals) or first-order Taylor approximations (Molchanov et al., 2017) to estimate the loss increase from weight removal. Recent work extends these ideas to attention heads (Michel et al., 2019; Voita et al., 2019) or attention matrices (He et al., 2025a). Several studies have investigated where factual knowledge is stored in transformers. Geva et al. (2021) show that FFN layers function as key-value memories. Meng et al. (2022) use causal tracing to locate factual associations in specific MLP layers. Our Fisher-importance-based attribution further provides a finer-grained lens for model attribution and understanding parameter specialization than layer-structure-wise analysis alone.

## B Critical Dimensions Coincide with FFN Activation Outliers

Section 2.2 establishes that zero-masking the top-12 Fisher-ranked intermediate dimensions of Qwen1.5-MoE collapses GSM8K from $35.9\%$ to $0.8\%$ while leaving multi-choice knowledge tasks largely intact. This appendix examines the forward-pass activation behaviour of those twelve dimensions and places them within the broader literature on activation outliers (Sun et al., 2024a). We make a deliberately limited claim: the critical dimensions are extreme upper-tail activation outliers that share the qualitative signature of – but are not at the canonical scale of – the massive-activation phenomenon documented in dense LLMs.

**Fisher and activation magnitude are not statistically independent.** Before reporting the activation statistics we make explicit a known mathematical dependency that prevents interpreting them as an independent confirmation of Fisher. For the down-projection slab, the gradient with respect to weight $(W_i^{\text{down}})_{k,j}$ factors as $\partial\mathcal{L}/\partial(W_i^{\text{down}})_{k,j} = (\partial\mathcal{L}/\partial h_k^{(\ell)}) \cdot a_j^{(i)}$, where $a_j^{(i)}$ is the post-activation along intermediate dimension $j$ of expert $i$ and $\partial\mathcal{L}/\partial h_k^{(\ell)}$ is the upstream gradient at the residual stream. Squaring and summing this term across $k$ – the form that enters the down-projection contribution of Eq. 5 – therefore scales with $(a_j^{(i)})^2$. High-activation channels accumulate Fisher mass through this term by construction, so any positive correlation between Fisher rank and activation magnitude is partly expected, not a separate signal. The gate- and up-projection contributions in Eq. 5 do not factor in the same way, so the coupling is partial rather than total, but it is strong enough that we treat the activation analysis below as *descriptive* – characterising the magnitude regime the top-12 occupy and connecting them to the activation-outlier literature – rather than as an independent attribution channel.

**Measurement.** For every routed expert in Qwen1.5-MoE-A2.7B and every intermediate dimension inside it (in total $\sim$2.03M dimensions across 24 MoE layers and 60 experts per layer), we record the mean absolute post-activation $|a_j|$ on the GSM8K calibration set (128 samples, the same set used to compute Fisher). We then compare the population distribution of $|a_j|$ against the values observed at the twelve dimensions whose Fisher score is largest.

| Rank | **mean** $\lvert a \rvert$ | $\times$ **pop. med.** | $\times$ **pop.** $p_{99}$ |
|---|---|---|---|
| *median* | 0.17 | 1× | 0.15× |
| #1 | 26.1 | 150× | 22× |
| #2 | 77.9 | 450× | 66× |
| #3 | 20.4 | 118× | 17× |
| #4 | 26.5 | 153× | 23× |
| #5 | 12.7 | 73× | 11× |
| #6 | 5.9 | 34× | 5× |
| #7 | 4.7 | 27× | 4× |
| #8 | 20.2 | 117× | 17× |
| #9 | 4.0 | 23× | 3× |
| #10 | 2.7 | 16× | 2× |
| #11 | 1.3 | 7× | 1× |
| #12 | 1.4 | 8× | 1× |

Table 7: Forward activation magnitudes of the top-12 Fisher-ranked intermediate dimensions in Qwen1.5-MoE on GSM8K.

**Population baseline is sharply skewed.** The empirical distribution of mean $|a|$ is heavy-tailed even before we look at any Fisher-selected outliers: median 0.17 and $p_{99} = 1.17$. In other words, fewer than 1% of all FFN intermediate dimensions exceed mean activation magnitude 1.17, and the typical dimension carries a forward signal an order of magnitude smaller.

Top-12 Fisher dimensions are extreme upper-tail outliers – but smaller than canonical massive activations. Table 7 reports, for each of the twelve dimensions, the mean $|a|$ together with its ratio to the population median and to the population $p_{99}$. The top nine dimensions carry mean $|a|$ between 4.0 and 77.9, i.e., 23× to 450× the population median and 3× to 66× the population $p_{99}$. Even the weakest two of the twelve (ranks 11 and 12) sit at 7–8× the median, comfortably above the 99th percentile of the entire MoE FFN. For calibration against the original literature, Sun et al. (2024a) report *massive activations* in dense LLMs whose magnitudes exceed the channel mean by roughly $10^4\times$ – two to three orders of magnitude beyond what we observe here. We therefore describe the top-12 as *extreme upper-tail activation outliers* that share the qualitative signature of the massive-activation phenomenon – a tiny subset of hidden coordinates carrying disproportionate forward-pass mass – without claiming they are massive activations in the strict numerical sense of the original definition. The MoE setting may also dilute the per-channel magnitude relative to a dense backbone, since each expert is activated only on a fraction of tokens.

The Spearman rank correlation between Fisher and mean $|a|$ over the twelve dimensions is 0.91. Given the gradient–activation coupling noted above, this correlation is the expected direction and approximately the expected strength: it confirms that intermediate-dimension Fisher concentrates on high-activation channels, but does not establish a statistically independent attribution channel. Read this way, the coupling is also the reason intermediate-dimension Fisher is a useful tractable attribution: by inheriting weight from the activation-driven down-projection term, it inherits an inductive bias toward exactly the outlier channels that prior work on activation outliers and attention sinks has shown to be load-bearing (Appendix C).

**Internal structure within the top-12.** Ranks 1–9 are extreme outliers deep in the upper tail of the population distribution ($\geq 23\times$ the median), while ranks 10–12 are still above $p_{99}$ but no longer of the same magnitude (7–16×). This tapering is consistent with activation outliers being concentrated in only a handful of channels per model – the first nine ranks carry the bulk of the abnormal forward-pass mass that, as Appendix C shows, is associated with the mid-stack attention sink.

## C Masking Critical Dimensions Reduces the Mid-Stack Attention Sink

### C.1 BOS Attention-Sink Analysis

Appendix B shows that the twelve Fisher-critical intermediate dimensions of Qwen1.5-MoE coincide with extreme upper-tail outliers in the FFN activation distribution – a milder MoE analogue of the massive activations documented in dense LLMs by Sun et al. (2024a). This appendix reports a second, downstream property of those dimensions: masking them substantially reduces the BOS attention sink (Xiao et al., 2024) in the mid-stack layers where the sink dominates. We use this observation to relate the accuracy dissociation in Table 1 to a known mechanistic concept, while being explicit about what the experiment can and cannot establish.

In a softmax attention layer the attention weights for each query must sum to one. When no key is genuinely relevant, optimisation pressure has been shown to push probability mass onto a low-information "parking" position – typically the BOS token (Xiao et al., 2024). This sink stabilises decoding in two ways: it absorbs unallocated probability mass without distorting on-topic tokens, and it provides a residual-stream anchor whose norm keeps mid-stack pre-softmax logits inside a numerically well-behaved range. Sun et al. (2024a) further argue that the sink is supported by extreme FFN activations that progressively inflate the BOS residual-stream norm across depth.

We measure, per decoder layer, the mean fraction of softmax attention paid to the BOS token, averaged across attention heads and across all positions of every GSM8K test prompt. We compare the base model against the Mask-Top-12 variant – the identical model with those twelve intermediate dimensions zeroed in the forward pass. Layers are grouped into bands that reflect the baseline sink profile of Qwen1.5-MoE.

| Layer band | base | Mask-Top-12 | drop |
|---|---|---|---|
| L0–1 | 0.07–0.09 | 0.07–0.09 | −0.00 |
| L2–6 | 0.30–0.38 | 0.22–0.35 | −0.05 |
| **L7–13** | **0.34–0.44** | **0.10–0.18** | **−0.24** |
| **L14–17** | 0.32–0.33 | 0.07–0.10 | **−0.25** |
| L18–21 | 0.17–0.22 | 0.02–0.03 | −0.17 |
| L22–23 | 0.09–0.11 | 0.03–0.04 | −0.06 |

Table 8: Mean attention to BOS per decoder-layer band on GSM8K with Qwen1.5-MoE-A2.7B.

Masking the top-12 substantially reduces the mid-stack sink. Table 8 shows that Mask-Top-12 attenuates the sink in the layers where it is strongest at baseline. In the mid-stack (L7–17) mean BOS attention falls from 0.32–0.44 down to 0.07–0.18, roughly a 70% reduction. The early sink-free layers (L0–1) are unchanged; the weaker early-middle and late tails (L2–6, L18–23) are only mildly attenuated. The layer-localised pattern is consistent with the twelve dimensions contributing meaningfully to the mid-stack sink rather than driving attention behaviour uniformly across depth.

We are deliberate about the strength of this claim. The result establishes a strong *association*: the twelve dimensions are activation outliers (Appendix B), they sit on the same family of channels prior work has linked to BOS attention dynamics (Sun et al., 2024a; Xiao et al., 2024), and removing them produces a sizeable mid-stack sink reduction. Appendix C.2 adds size-matched controls for activation magnitude, layer location, and random selection. We treat the result as evidence that the critical dimensions identified by Fisher are part of – not necessarily the entirety of – the FFN-side substrate that maintains the mid-stack sink.

Two controls partially tighten the attribution. (i) Removing the universally-critical core $\mathcal{K}_\cap$ ($\sim$4.88% of all dimensions; §2.3) – which strictly contains the top-12 – collapses BOS attention globally rather than only in the mid-stack, consistent with the larger set containing additional sink-supporting dimensions beyond the top-12. (ii) Removing the comparably-sized but Fisher-redundant set $\mathcal{D}_\cap$ ($\sim$4.01%) leaves BOS attention essentially unchanged in every layer band. Together with the size-matched controls in Appendix C.2, these results argue against generic ablation, activation magnitude, or mid-stack location as sufficient explanations.

**Linking sink reduction to the accuracy dissociation.** The pattern of sink reduction lines up with the accuracy pattern in Table 1: generation-heavy benchmarks (`GSM8K`, `MATH`, `HumanEval`, `MBPP`) require many autoregressive decoding steps, each of which depends on the numerically stable attention dynamics that the mid-stack sink maintains; once the sink is attenuated, long-form generation degenerates into the echoes and garbled outputs catalogued in Appendix J (31.1% echo, 7.5% empty – behaviours absent at baseline). Short-answer MCQ tasks (`MMLU`, `CEval`, `CMMLU`, `BBH`) require only a

single-token prediction conditioned on the prompt, do not heavily exercise long-range attention dynamics, and correspondingly retain 90–98% of base accuracy. We frame this as a coherent mechanistic explanation: the sink-reduction direction and the accuracy-collapse direction match, and that match is consistent with the literature linking outlier FFN activations to attention sinks.

**Summary.** Taken together, Appendices B and C place the twelve Fisher-critical dimensions in a coherent mechanistic context: **(i)** intermediate-dimension Fisher localises capability onto a tiny set of channels whose Fisher scores are $\sim 1000\times$ the population mean; **(ii)** those channels sit in the extreme upper tail of the FFN activation distribution, a milder MoE analogue of the massive-activation phenomenon documented for dense LLMs; **(iii)** masking them substantially reduces the mid-stack BOS attention sink while leaving sink-free layers alone, mirroring the generation-vs-MCQ dissociation in Table 1. We present this as a consistent pattern that connects intermediate-dimension Fisher to known mechanistic concepts.

### C.2 Size-Matched Controls for Fisher-Critical Dimensions

We mask 12 dimensions while controlling set size, activation magnitude, and layer location. Only the Fisher-critical set collapses `GSM8K` and suppresses the mid-stack BOS attention sink; matched controls remain near the base model (Table 9). This supports a Fisher-specific sensitivity pattern rather than a generic outlier or layer-location effect.

## D Detailed Experimental Results

This appendix collects the full per-benchmark numbers that underlie the figures and tables in the main paper. Each block below corresponds to one main-paper figure or table.

| Setting (12 dimensions unless noted) | Mean BOS attention (L7–17) | GSM8K | MMLU |
|---|---|---|---|
| Base model | 0.32–0.44 | 35.9 | 59.3 |
| Mask top-12 Fisher-critical | 0.128 | 0.8 | 53.4 |
| Mask matched $\lvert\text{activation}\rvert$ (non-Fisher) | 0.359 | 33.0 | 59.0 |
| Mask same mid-stack layers (non-Fisher) | 0.361 | 33.6 | 58.9 |
| Mask random (3 seeds) | 0.359 | 32.7–33.0 | 59.0–59.1 |
| Mask redundant set | 0.359 | 33.1 | 59.2 |

Table 9: Size-matched masking controls on Qwen1.5-MoE-A2.7B. Accuracy is reported in percent; non-Fisher controls additionally match activation magnitude or layer location where indicated.

| Method | MMLU | HumanEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH |
|---|---|---|---|---|---|---|---|---|
| Base model | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 |
| *Existing Importance Metrics* | | | | | | | | |
| Activation-based | 26.3 | 0.0 | 0.0 | 42.5 | 43.1 | 0.0 | 1.9 | 24.5 |
| Score-based | 37.2 | 0.0 | 0.8 | 13.5 | 32.3 | 0.1 | 1.9 | 23.5 |
| Magnitude-based | 18.6 | 0.8 | 0.0 | 33.3 | 35.8 | 0.1 | 1.7 | 13.2 |
| *Proposed Fisher Importance* | | | | | | | | |
| **Router-level Fisher** | **39.4** | 12.9 | 8.1 | **48.8** | **47.7** | 3.5 | 16.9 | **25.7** |
| **Expert-level Fisher** | 37.9 | **12.9** | **11.7** | 48.6 | 44.2 | **3.9** | **18.0** | 24.1 |

Table 10: Raw data underlying **Figure 2**: expert-level pruning with existing importance metrics on Qwen1.5-MoE at $p = 50\%$ MoE compression ratio.

| Setting | MMLU | HumanEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH |
|---|---|---|---|---|---|---|---|---|
| Base model | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 |
| *Top-masking* | | | | | | | | |
| Mask-Top-12 | 53.4 | 4.3 | 17.0 | 57.4 | 59.3 | 1.1 | 0.8 | 23.6 |
| *Critical vs Redundant Dimension Removal* | | | | | | | | |
| Remove redundant ($\mathcal{D}_\cap$) | 58.8 | 33.3 | 23.6 | 62.9 | 61.6 | 8.7 | 31.2 | 31.2 |
| Remove critical ($\mathcal{K}_\cap$) | 5.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 |

Table 11: Raw data underlying **Table 1** (mask-top-12 and critical/redundant dimension removal). Performance (%) on Qwen1.5-MoE-A2.7B.

| Method | MMLU | HumanEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH |
|---|---|---|---|---|---|---|---|---|
| Base (no pruning) | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 |
| *Drop 25%* | | | | | | | | |
| MoE-Pruner | 53.9 | 12.2 | 0.8 | 63.5 | 66.0 | 3.8 | 7.0 | 22.6 |
| MoE compression | 54.7 | 20.7 | 4.8 | 62.0 | 62.2 | 1.9 | 11.5 | 27.8 |
| MoE compression (activation) | 54.2 | 1.8 | 10.4 | 61.2 | 58.9 | 1.2 | 4.6 | 29.0 |
| **Fisher-IntDim-E (ours)** | **56.1** | **30.5** | **24.3** | **69.7** | **67.4** | **11.2** | **37.3** | **29.5** |
| *Drop 50%* | | | | | | | | |
| MoE-Pruner | 18.6 | 0.8 | 0.0 | 33.3 | 35.8 | 0.1 | 1.7 | 13.2 |
| MoE compression | 37.2 | 0.0 | 0.8 | 13.5 | 32.3 | 0.0 | 1.9 | 23.5 |
| MoE compression (activation) | 26.3 | 0.0 | 0.0 | 42.5 | 43.1 | 0.0 | 1.9 | 24.5 |
| **Fisher-IntDim-E (ours)** | **50.3** | **21.2** | **20.5** | **62.0** | **61.6** | **8.0** | **35.0** | **28.2** |
| *Drop 75%* | | | | | | | | |
| MoE-Pruner | 4.8 | 0.0 | 0.0 | 13.7 | 16.1 | 0.1 | 0.0 | 3.7 |
| MoE compression | 20.6 | 0.0 | 0.0 | 1.6 | 15.3 | 0.3 | 0.7 | 5.3 |
| MoE compression (activation) | 23.0 | 0.0 | 0.0 | 11.5 | 15.3 | 1.0 | 0.7 | 14.2 |
| **Fisher-IntDim-E (ours)** | **42.2** | **15.2** | **11.9** | **49.7** | **49.0** | **3.6** | **27.0** | **26.4** |

Table 12: Raw data underlying **Figure 3**: Qwen1.5-MoE accuracy (%) under expert-level compression at three drop ratios.

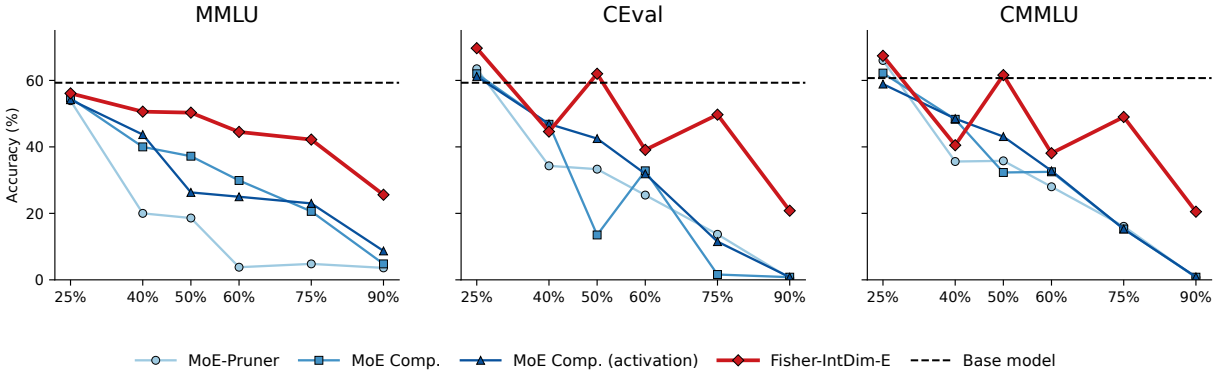


Figure 6: Fine-grained compression-ratio sweep on the knowledge benchmarks `MMLU`, `CEval`, and `CMMLU`. We extend the ratios in Fig. 4 with compression ratios of 40%, 60%, and 90%. Across all three benchmarks, Fisher-IntDim-E remains substantially more robust than expert-level baselines as compression becomes more aggressive. Dashed lines denote the uncompressed base model.

| **Method** | **MMLU** | **HumanEval** | **MBPP** | **CEval** | **CMMLU** | **MATH** | **GSM8K** | **BBH** |
|---|---|---|---|---|---|---|---|---|
| Base model | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 |
| *Expert-level compression (Fisher-based)* | | | | | | | | |
| MoE compression (Fisher) | 39.4 | 12.9 | 8.1 | 48.8 | 47.7 | 3.5 | 16.9 | 25.7 |
| Expert-level Fisher | 37.9 | 12.9 | 11.7 | 48.6 | 44.2 | 3.9 | 18.0 | 24.1 |
| *Intermediate dimension compression (Fisher-based)* | | | | | | | | |
| **IntDim-E (ours)** | **50.3** | **21.2** | **20.5** | **62.0** | **61.6** | **8.0** | **35.0** | **28.2** |

Table 13: Raw data underlying **Table 2**: expert-level vs. intermediate dimension compression at $p = 50\%$ on Qwen1.5-MoE-A2.7B.

| **Granularity** | **Metric** | **Knowledge** | | | **Coding** | | **Math** | | | **Reas.** | **Avg.** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **MMLU** | **CEval** | **CMMLU** | **HE** | **MBPP** | **MATH** | **GSM8K** | **MR** | **BBH** | |
| – | Base (no compression) | 59.3 | 59.3 | 60.7 | 32.6 | 24.6 | 13.0 | 35.9 | 76.7 | 30.7 | **43.6** |
| Expert | Random | 35.8 | 35.1 | 37.1 | 3.3 | 0.7 | 0.6 | 2.6 | 1.7 | 16.1 | **14.8** |
| | Activation ratio | 26.3 | 42.5 | 43.1 | 0.0 | 0.0 | 0.0 | 1.9 | 3.3 | 24.5 | **15.7** |
| | Router score | 37.2 | 13.5 | 32.3 | 0.0 | 0.8 | 0.0 | 1.9 | 2.0 | 23.5 | **12.4** |
| | Weight magnitude | 18.6 | 33.3 | 35.8 | 0.8 | 0.0 | 0.1 | 1.7 | 2.7 | 13.2 | **11.8** |
| | Fisher | 37.9 | 48.6 | 44.2 | 12.9 | 11.7 | 2.4 | 18.0 | 37.7 | 24.1 | **26.4** |
| IntDim-E | Random | 47.4 | 45.1 | 46.9 | 6.7 | 1.1 | 1.5 | 3.2 | 7.2 | 21.2 | **20.0** |
| | Weight magnitude | 49.3 | 51.6 | 54.0 | 14.6 | 5.6 | 2.2 | 7.5 | 9.3 | 26.7 | **24.5** |
| | Activation ratio | 49.4 | 50.4 | 50.7 | 7.9 | 3.3 | 7.6 | 29.8 | 78.2 | 27.2 | **33.8** |
| | **Fisher (ours)** | **50.3** | **62.0** | **61.6** | **21.2** | **20.5** | **8.0** | **35.0** | **61.5** | **28.2** | **38.7** |

Table 14: Granularity $\times$ metric factorial on Qwen1.5-MoE-A2.7B at $p = 50\%$; per-benchmark accuracy (%). Rows are the (granularity, metric) cells. expert-random is the mean of three seeds. All remaining cells reuse previously reported measurements, with the same evaluator, calibration data, and per-task decoding settings throughout. "HE" denotes HumanEval and "MR" denotes MultiArith.

| **Method (Qwen1.5-MoE-A2.7B)** | **In-domain** | | | **Out-of-domain** | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **GSM8K (Calib.)** | **MATH** | **MultiArith** | **MMLU** | **HumanEval** | **MBPP** | **CEval** | **CMMLU** | **BBH** |
| MoE-Pruner | 1.7 | 0.0 | 1.8 | 18.6 | 0.6 | 0.0 | 33.3 | 35.8 | 13.2 |
| MoE comp. | 1.9 | 0.3 | 0.3 | 33.4 | 1.2 | 0.0 | 37.8 | 39.8 | 22.1 |
| MoE comp. (activation) | 1.9 | 0.2 | 0.8 | 33.7 | 0.0 | 0.3 | 40.2 | 40.4 | 12.3 |
| MoE comp. (Fisher) | 16.9 | 0.7 | 26.2 | 36.1 | 7.9 | 0.5 | 38.0 | 36.8 | 23.0 |
| Expert-level Fisher | 18.0 | 0.7 | 31.7 | 36.5 | 7.3 | 4.3 | 40.2 | 39.4 | 24.6 |
| **Fisher-IntDim-E (ours)** | **35.0** | **4.7** | **70.8** | **48.1** | 5.5 | **5.6** | **50.6** | **49.4** | 24.2 |

Table 15: Raw data underlying **Table 7**: strict zero-shot evaluation on Qwen1.5-MoE-A2.7B. GSM8K is calibration and is part of the in-domain set.

| Method | Disk | VRAM | MMLU | HumanEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH | MultiArith | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Base Model | 26.7 | 58.1 | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 | 76.7 | 43.6 |
| Base Model + AWQ | 7.9 | 8.6 | 53.8 | 20.7 | 4.8 | 49.0 | 47.9 | 4.5 | 16.1 | 22.0 | 14.3 | 25.9 |
| Fisher-IntDim-E | 15.1 | 30.1 | 50.3 | 21.2 | 20.5 | 62.0 | 61.6 | 8.0 | 35.0 | 28.2 | 61.5 | 38.7 |
| **Fisher-IntDim-E + AWQ** | 5.0 | 6.5 | 46.3 | 12.2 | 14.4 | 52.8 | 51.5 | 2.9 | 12.4 | 24.6 | 25.2 | 26.9 |

Table 16: Raw data underlying **Table 8**: stacking 4-bit AWQ on top of Fisher-MoE at MoE compression ratio $p = 50\%$ on Qwen1.5-MoE-A2.7B. Disk and VRAM are in GiB.

| Calibration for one compressed model | MMLU | CEval | CMMLU | HEval | MBPP | MATH | BBH | Avg. |
|---|---|---|---|---|---|---|---|---|
| Base, uncompressed | 59.3 | 59.3 | 60.7 | 32.6 | 24.6 | 13.0 | 30.7 | 40.0 |
| GSM8K-only calibration | 48.1 | 50.6 | 49.4 | 5.5 | 5.6 | 4.7 | 24.2 | 26.9 |
| Mixed math/code/knowledge/reasoning | 51.1 | 56.8 | 58.0 | 18.9 | 15.4 | 6.6 | 27.5 | 33.5 |

Table 20: Mixed-domain calibration on Qwen1.5-MoE-A2.7B at $p = 50\%$. We distinguish same-domain/same-format, same-domain/different-format, and different-domain/different-format calibration, with all compared methods using the same calibration data. Table 15 provides a deliberately narrow math-only transfer test, in which code is the most distant domain. For broader deployment, mixed calibration improves coverage across all task families. The mixed set contains 128 samples, with 32 samples from each of math, code, knowledge, and reasoning; the same compressed model is evaluated on every benchmark.

| Batch-size sweep (input 1024, output 256; tokens/s) | | | | | |
|---|---|---|---|---|---|
| **Batch** | **1** | **8** | **32** | **64** | **128** |
| Base | 369 | 2,113 | 5,978 | 9,257 | 13,093 |
| IntDim-E | 400 | 2,386 | 7,295 | 11,002 | 17,103 |
| Speedup | 1.08× | 1.13× | 1.22× | 1.19× | 1.31× |

| Input-length sweep (batch 128, output 256) | | | | |
|---|---|---|---|---|
| **Input length** | **128** | **512** | **1024** | **2048** |
| Speedup | 1.15× | 1.25× | 1.31× | 1.28× |

| Output-length sweep (batch 128, input 1024) | | |
|---|---|---|
| **Output length** | **64** | **256** |
| Speedup | 1.23× | 1.31× |

Table 21: Expanded OLMoE-1B-7B serving results on H100 across five batch sizes, four input lengths, and two output lengths. Speedups persist across all tested configurations and generally increase with larger workloads; evaluation on other GPU platforms remains future work.

| Method (Fisher signal, $p = 50\%$) | Unit | Shrinks? | MMLU | CEval | CMMLU | HEval | MBPP | MATH | GSM8K | MArith | BBH | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fisher-Wanda (2:4) | weight, 2:4 | no | 14.1 | 6.8 | 5.0 | 0.0 | 0.0 | 0.2 | 2.0 | 4.3 | 4.9 | 4.1 |
| Fisher-Wanda (unstructured 50%) | weight | no | 18.1 | 18.8 | 22.6 | 0.0 | 0.0 | 0.2 | 2.6 | 10.0 | 10.7 | 9.2 |
| Fisher-SparseGPT (2:4 + OBS) | weight, 2:4+comp. | no | 23.0 | 25.1 | 26.6 | 0.0 | 0.0 | 0.3 | 9.2 | 49.2 | 13.6 | 16.3 |
| Expert-level Fisher | whole expert | yes | 37.9 | 48.6 | 44.2 | 12.9 | 11.7 | 2.4 | 18.0 | 37.7 | 24.1 | 26.4 |
| **Fisher-IntDim-E** | intermediate dim. | yes | **50.3** | **62.0** | **61.6** | **21.2** | **20.5** | **8.0** | **35.0** | **61.5** | **28.2** | **38.7** |

Table 22: Fixed-Fisher granularity ladder on Qwen1.5-MoE-A2.7B at $p = 50\%$. Holding empirical Fisher and all other settings fixed, we compare per-weight, whole-expert, and intermediate-dimension pruning units. Fisher-IntDim-E performs best and physically shrinks dense FFNs, whereas the weight-level controls retain sparse matrix shapes. Every row uses GSM8K-128 calibration and the same empirical Fisher signal; only the pruning unit changes.

## E Additional Experimental Settings

**Model Architecture and Dataset:** In our experimental setup, we use open-weight OLMoE-1B-7B-0125 (Muennighoff et al., 2024), Qwen1.5-MoE-A2.7B (Yang et al., 2024a), Qwen3-30B-A3B (Yang et al., 2025), and Qwen3.5-35B-A3B (Qwen Team, 2026) to conduct experiments. We adopt two data configurations: a general task configuration following technical reports (Yang et al., 2024a; Qwen Team, 2024) for the lightweight pre-trained LLMs (Qwen1.5-MoE-A2.7B, OLMoE-1B-7B-0125), and a long CoT configuration for the stronger and larger reasoning LLMs (Qwen3-30B-A3B, Qwen3.5-35B-A3B). For Qwen1.5-MoE-A2.7B and OLMoE-1B-7B-0125, we evaluate on `GSM_8K` (Cobbe et al., 2021), `MMLU` (Hendrycks et al., 2021a), `HumanEval` (Chen et al., 2021), `MBPP` (Austin et al., 2021), `CEval`, `CMMLU` (Li et al., 2024), `MATH` (Hendrycks et al., 2021b), `BBH` (Suzgun et al., 2023), and `MultiArith` (Roy and Roth, 2015), which together cover knowledge, code generation, mathematical reasoning, and general reasoning. For Qwen3-30B-A3B and Qwen3.5-35B-A3B, we evaluate on the SOTA long-CoT math reasoning benchmarks `AIME2026`, `AIME2025`, `GPQA-Diamond` (Rein et al., 2024), `MATH-500`, and `Olympiad Bench`.

All pretrained backbones (Qwen1.5-MoE, Qwen3/Qwen3.5, OLMoE) and all evaluation benchmarks used in this paper are public research artifacts; we use each under its original license (Qwen / Apache-2.0 for the Qwen family, Apache-2.0 for OLMoE, and the respective research licenses for `GSM8K`, `MATH`, `MMLU`, `HumanEval`, `MBPP`, `CEval`, `CMMLU`, `BBH`, `MultiArith`, `AIME`, `GPQA-D`, `Olympiad Bench`, `MATH-500`, and `Stanford-S1`). Our released code and checkpoints inherit the license of the corresponding base model.

**MoE Compression Baselines:** For state-of-the-art (SOTA) MoE compression baselines, we include the unified MoE compression framework of (He et al., 2025b) (denoted *MoE compression*, which uses router scores as the importance metric), and MoE-Pruner (Xie et al., 2024) (weight magnitude). All these baselines operate at the expert granularity. To more comprehensively compare importance signals while holding the compression framework fixed, we further extend the unified framework with two alternative importance metrics (activation ratio and router-Fisher) yielding two controlled variants that we denote *MoE compression (activation)* and *MoE compression (Fisher)*. This setup lets us attribute performance differences to the importance metric and granularity.

**Fisher-MoE Variants.** We propose four Fisher-MoE variants that differ in pruning granularity and allocation flexibility. **(1) Expert-level Fisher** treats each expert as the pruning unit, where the Fisher score of expert $i$ is computed over the union of its parameters $W_i^{\text{gate}}$, $W_i^{\text{up}}$, and $W_i^{\text{down}}$, and entire experts are removed according to this score. **(2) IntDim-E** applies Fisher scoring at the intermediate dimension level but keeps the top $(1-p)$ fraction of dimensions within each expert, enforcing the same compression ratio for every expert. **(3) IntDim-L** pools intermediate dimensions across all experts within the same MoE layer and keeps the top $(1-p)$ fraction per layer, allowing the retained dimensions to be redistributed among experts in that layer. **(4) IntDim-G** pools intermediate dimensions across the entire model and keeps the top $(1-p)$ fraction globally, providing the most flexible allocation across both layers and experts. All intermediate dimension variants use the same Fisher scoring criterion and overall parameter budget, allowing us to isolate the effect of increasingly flexible fine-grained allocation.

**Calibration–Evaluation Disjointness:** For every benchmark used as both a calibration source and an evaluation target (e.g., GSM8K, MATH, MultiArith, MBPP, HumanEval), we use the standard training split for calibration (and SFT, where applicable) and evaluate on the standard test split, which has no overlap with calibration. The same disjoint train/test convention is used for every other in-domain calibration–evaluation pair throughout the paper. To ensure fair comparison, all methods use the same calibration data to compute importance scores and perform compression.

**Training Framework and Hyper-parameters:** We use the `huggingface-trl` (von Werra et al., 2020) library with ZeRO-2 or ZeRO-3 (Ren et al., 2021) for fine-tuning, and `vllm` (Kwon et al., 2023), `lighteval` (Habib et al., 2023), and `accelerate` (Gugger et al., 2022) for inference and evaluation. Both training and evaluation use bf16.

**Computational Resources:** We run all experiments, baseline implementations, and post-training on 8×NVIDIA H100 80 GB GPUs or 8×NVIDIA

A100 80 GB GPUs. CPU–GPU communication is over PCIe Gen4, and inter-GPU communication is over NVLink-3.

## F Ablation Study on Calibration Set Size

The Fisher importance score (§2.1) is a Monte-Carlo square of $|\nabla_W \mathcal{L}(x, y)|$ over a calibration set $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$:

$$s^{\text{Fisher}}(W) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} |\nabla_W \mathcal{L}(x, y)|^2 , \qquad N = |\mathcal{D}|. \tag{9}$$

Because every $(x, y)$ requires a forward and a backward pass through the full MoE, the cost of computing $s^{\text{Fisher}}$ scales linearly with $N$. The natural question is how large $N$ must be for the resulting Fisher ranking—and thus the compressed model that ranking selects—to stabilize.

We answer this empirically: we fix the backbone (Qwen1.5-MoE-A2.7B), the MoE compression ratio ($p = 50\%$ via Fisher-IntDim-E), the calibration domain (domain-matched calibration data), and the decoding settings (greedy, vLLM v0.8.4, bf16), then vary only $N \in \{32, 64, 128, 256, 512\}$. Table 23 reports downstream accuracy of the resulting compressed model. The 'N/A' results for $N = 512$ occur because some datasets contain fewer than 512 samples for calibration.

Table 23: Fisher-IntDim-E calibration-size sweep on Qwen1.5-MoE-A2.7B at $p = 50\%$ (domain-matched calibration, greedy decoding via vLLM). Each column reports downstream accuracy of the compressed model produced when the Fisher score is estimated from $N$ calibration samples. Higher is better.

| Task | N=16 | N=32 | N=64 | N=128 | N=256 | N=512 |
|---|---|---|---|---|---|---|
| GSM8K | 31.6 | 33.7 | 38.5 | 35.0 | 40.3 | 40.9 |
| MMLU | 50.3 | 50.4 | 50.3 | 50.3 | 50.0 | 53.0 |
| CEval | 59.3 | 57.5 | 61.1 | 62.0 | 62.4 | N/A |
| CMMLU | 58.9 | 59.7 | 60.9 | 61.6 | 60.7 | N/A |
| MATH | 6.8 | 5.1 | 7.7 | 8.0 | 7.1 | 7.8 |

Three observations stand out. First, the Fisher ranking is sample-efficient: at $N = 32$ the resulting compressed model already retains 33.7% on GSM8K and 50.4% on MMLU, both above the strongest expert-level Fisher baseline at $N = 128$ (Table 3, *Fisher-Expert*: 18.0/37.9). This is consistent with the empirical Fisher being a Monte-Carlo estimator of $\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{cal}}}[|\nabla_W \mathcal{L}|]$: the average is dominated by a small number of high-gradient examples, so even tens of samples concentrate the ranking around the right tail. Second, the curve is monotone but flat past $N = 128$: MMLU and MATH show marginal improvement from $N = 128$ to $N = 512$ (50.3% $\rightarrow$ 53.0% and 8.0 $\rightarrow$ 7.8) despite a $4\times$ increase in compute.

**Default.** Following the knee of the curve, we use $N = 128$ for all main-paper experiments. This setting is the smallest sample budget for which the Fisher-IntDim-E model is within $\sim$2 points of the $N = 512$ result on MMLU and MATH, while requiring only 128 forward+backward passes per backbone, a cost within one minute on one H100 node even for the 30B-parameter MoEs.

## G AWQ Quantization Settings

**Quantization precision.** AWQ is configured with per-group W4A16-asymmetric quantization: weight bit-width `w_bit=4` (INT4), `zero_point=True` (asymmetric, per-group scale and zero-point), group size `q_group_size=128`, activations kept in the original FP16/BF16 dtype, GEMM kernel format, and packed INT4 `safetensors` output.

**Calibration.** PileVal (alternatives: C4, WikiText2), `split=train`, text column `text`, at most 128 calibration samples (script variants: 32, 1024), maximum sequence length 512 (variant: 256), duo-scaling enabled (jointly optimize the scale w.r.t. weight and activation).

**Quantized layers.** Quantization is applied per `Qwen2MoeDecoderLayer`; within each layer, weights are partitioned into *scaling groups* that share AWQ smoothing and quantization jointly.

- **Attention.** (i) Input LayerNorm → [Q-proj, K-proj, V-proj] (grouped scaling); (ii) V-proj → [O-proj] (only when `num_kv_heads == num_heads`; O-proj is still quantized).
- **MoE sparse layers** (most decoder layers). (i) Post-Attention LayerNorm → [all expert gate/up projections + shared-expert gate/up] (jointly scaled across 60 experts × 2 + 2 shared modules); (ii) per expert: up-proj → [down-proj]; (iii) shared expert: up-proj → [down-proj].
- **Dense MLP layers** (`mlp_only_layers`). (i) Post-Attention LayerNorm → [MLP gate-proj, up-proj]; (ii) MLP up-proj → [MLP down-proj].

**Modules excluded from quantization.** `modules_to_not_convert = ["gate", "shared_expert_gate"]`: the MoE router (`mlp.gate`, computes routing logits) and the shared-expert gate (`shared_expert_gate`, sigmoid gating). AutoAWQ defaults additionally skip `lm_head`, all embedding layers, and every LayerNorm/RMSNorm.

**Additional behaviours.** `fuse_layers` is a no-op for Qwen2-MoE (no QKV/MLP fusion; module structure is preserved). `get_act_for_scaling` uses `is_scalable=False` (no extra scalable activations at the decoder-layer level). `move_embed` transfers embedding tokens to the configured device for calibration hidden-state collection. Layer type: `Qwen2MoeDecoderLayer`; sequence-length key: `max_position_embeddings`.

**Summary.** AWQ performs per-group W4A16-asymmetric quantization (group size 128) across Qwen2-MoE, covering all QKV/O projections in attention and gate/up/down modules in all routed and shared experts, while skipping routers (`mlp.gate`), the shared-expert gate, layer norms, embeddings, and the head. Calibration uses PileVal with 128 samples × 512 tokens by default and duo-scaling enabled; the output uses the GEMM kernel format.

## H Shared Dimensions Across Tasks

To further characterize how retained dimensions are shared across tasks, we compute pairwise overlaps. For each pair $(a, b)$, we define

$$\mathrm{Overlap}(a,b) \;=\; \frac{|\mathcal{K}_a \cap \mathcal{K}_b|}{|\mathcal{K}_a|} \;=\; \frac{|\mathcal{K}_a \cap \mathcal{K}_b|}{|\mathcal{K}_b|}, \tag{10}$$

where the equality holds since $|\mathcal{K}_a| = |\mathcal{K}_b|$ under a shared drop ratio. Table 24 shows the resulting overlap matrix. Three structured patterns emerge: **(1) Linguistic affinity.** The Chinese benchmarks C-EVAL and CMMLU exhibit the highest overlap (69.6%).**(2) Domain affinity.** Coding tasks (HUMANEVAL/MBPP, 66.5%) and math tasks (GSM8K/MATH, 65.3%) form similarly strong clusters.

Table 24: Pairwise overlap of kept intermediate dimensions between tasks. Values above 65% are highlighted in bold.

| | BBH | C-EVAL | CMMLU | GSM | HEVAL | MATH | MBPP | MMLU |
|---|---|---|---|---|---|---|---|---|
| BBH | 100.0 | 54.7 | 55.1 | 60.9 | 61.2 | 60.4 | 60.4 | 60.5 |
| C-EVAL | 54.7 | 100.0 | **69.6** | 54.9 | 55.6 | 56.9 | 54.5 | 58.3 |
| CMMLU | 55.1 | **69.6** | 100.0 | 53.8 | 54.2 | 55.4 | 53.6 | 57.6 |
| GSM | 60.9 | 54.9 | 53.8 | 100.0 | 59.3 | **65.3** | 58.0 | 59.7 |
| HEVAL | 61.2 | 55.6 | 54.2 | 59.3 | 100.0 | 62.5 | **66.5** | 60.6 |
| MATH | 60.4 | 56.9 | 55.4 | **65.3** | 62.5 | 100.0 | 60.5 | 64.0 |
| MBPP | 60.4 | 54.5 | 53.6 | 58.0 | **66.5** | 60.5 | 100.0 | 59.1 |
| MMLU | 60.5 | 58.3 | 57.6 | 59.7 | 60.6 | 64.0 | 59.1 | 100.0 |

# I Commonsense Benchmarks Exhibit High Variance Under Random Compression

In Section 1, we argue that an overlooked issue is that prior works predominantly evaluate compressed MoE models on commonsense reasoning benchmarks (e.g., ARC, PIQA, HellaSwag). We find that these benchmarks exhibit large variance across runs and are unreliable indicators of compression quality. For instance, randomly removing 50% of experts, performance on individual commonsense tasks can fluctuate by over 20 percentage points across different random seeds, and removing more parameters paradoxically yields better scores.

We compress Qwen1.5-MoE-A2.7B at 50% expert removal using two different random seeds and evaluate on eight commonsense reasoning tasks. No calibration data or importance metric is used—experts are selected uniformly at random. For comparison, we also include Fisher-based attention head compression at 25% and 50% drop ratios. Table 25 reports the results alongside the uncompressed base model.

Three phenomena are noteworthy:

**Extreme inter-seed variance.** The average commonsense accuracy differs by 18.4 percentage points between the two random seeds (9.1% vs. 27.5%). On individual tasks, the variance is even larger: BoolQ fluctuates by 47.0 points (12.0% vs. 59.0%) and PIQA by 33.6 points (8.6% vs. 42.2%). This level of variance means that any single-run comparison between compression methods on these benchmarks is essentially uninformative—the difference between "method A beats method B" and the reverse can be determined entirely by which random subset of experts happens to be removed.

**More compression can paradoxically improve scores.** Consider the attention head results: removing 50% of heads with Fisher-based selection achieves *higher* average commonsense accuracy (23.1%) than removing only 25% of heads (18.7%). On 6 out of 8 individual tasks—including ARC-C (16.0 vs. 11.5), HellaSwag (18.6 vs. 3.1), OBQA (21.4 vs. 17.0), PIQA (39.0 vs. 34.8), SIQA (26.4 vs. 8.4), and WinoGrande (32.8 vs. 22.2)—the more aggressively compressed model outperforms the less compressed one. This is nonsensical: removing more parameters should not improve the model. The explanation is that commonsense benchmarks have high random baselines (BoolQ: 50%, PIQA: 50% for binary/two-option tasks) and at high compression ratios the scores are dominated by noise rather than genuine model capability.

**Random removal can beat Fisher-guided compression.** Random seed 42 achieves higher commonsense accuracy (27.5%) than Fisher-based 25% attention head compression (18.7%), despite using no importance metric whatsoever and removing twice as many parameters. On BoolQ specifically, random removal scores 59.0%—above the 50% random baseline and far above the Fisher-guided 39.0%. This further confirms that commonsense benchmarks cannot reliably distinguish between compression strategies at moderate-to-high compression ratios.

These observations motivate our use of more challenging benchmarks (GSM8K, HumanEval, MMLU, MATH, BBH, CEval, CMMLU, MBPP) in the main paper, which have much lower random baselines and require genuine multi-step reasoning or generation.

# J Failure-Mode Dissection of GSM8K Responses at 0.0006% Critical Intermediate Dimension Removal

Section 2.2 reports that masking the top $\sim$12 most Fisher-important intermediate dimensions (0.0006% of the 2.03M MoE FFN intermediate dimensions in Qwen1.5-MoE-A2.7B) collapses GSM8K accuracy from 35.9% to 0.8%. To understand *how* the model fails rather than merely *that* it fails, we manually inspect all 1,319 GSM8K test outputs and classify each into one of eight categories. We compare the resulting distribution to the base model, where accuracy is 35.9% and the model still produces coherent multi-step reasoning. Decoding uses temperature 0.1 and `max_tokens`$=500$ in both settings.

**Categories.** We define eight mutually exclusive categories spanning the observed output behaviors: **Real reasoning:** a multi-step chain-of-thought with explicit arithmetic and connectives that reaches a numeric conclusion. **Partial reasoning:** reasoning tokens ("total", "step", "therefore") appear but the chain is incomplete or incoherent. **Exact echo:** a verbatim copy of the input question. **Truncated echo:** a partial copy of the question that loops or terminates mid-sentence. **Garbled echo:** the question text with name substitutions, dropped clauses, or reordered phrases (a semantics-

Table 25: Commonsense reasoning performance under random 50% expert removal (two seeds) and Fisher-based attention head compression on Qwen1.5-MoE-A2.7B. The base model (no compression) is shown for reference.

| **Method** | **ARC-C** | **ARC-E** | **BoolQ** | **HellaS** | **OBQA** | **PIQA** | **SIQA** | **WinoG** | **Avg.** |
|---|---|---|---|---|---|---|---|---|---|
| Base model | 45.3 | 61.1 | 74.8 | 35.5 | 41.0 | 69.1 | 50.8 | 51.0 | 53.6 |
| *Fisher-based attention head compression* | | | | | | | | | |
| Attn head 25% drop | 11.5 | 13.5 | 39.0 | 3.1 | 17.0 | 34.8 | 8.4 | 22.2 | 18.7 |
| Attn head 50% drop | 16.0 | 18.3 | 12.1 | 18.6 | 21.4 | 39.0 | 26.4 | 32.8 | 23.1 |
| *Random expert removal (no importance metric)* | | | | | | | | | |
| Random seed 123 | 9.0 | 9.1 | 12.0 | 4.1 | 11.8 | 8.6 | 9.0 | 9.2 | 9.1 |
| Random seed 42 | 17.8 | 20.5 | 59.0 | 21.0 | 22.4 | 42.2 | 17.9 | 19.5 | 27.5 |
| $\lvert\Delta\rvert$ (seed 123 vs. 42) | 8.8 | 11.4 | **47.0** | 16.9 | 10.6 | **33.6** | 8.9 | 10.3 | 18.4 |

free transformation). **Numeric only:** a bare digit, almost always "1". **Empty output:** no tokens generated. **Other:** short fragments, paraphrases, or otherwise garbled text not covered above.

**Distribution.** Table 26 reports the category breakdown at top-12 removal alongside the base model. Three patterns are noteworthy.

*(1) Reasoning collapses by an order of magnitude.* Real reasoning drops from 60.0% to 4.2% (a $14\times$ reduction) and partial reasoning drops from 7.6% to 2.1%. The 10 "correct" answers at top-12 removal are coincidental matches between numbers appearing in the echoed question text and the gold answer; none reflect residual computation. This reduction in reasoning capacity, rather than the raw accuracy drop, is the more direct signal that the removed dimensions support multi-step generation.

Table 26: Distribution of GSM8K output categories (1,319 test items) under base model and top-12 (0.0006%) reverse intermediate dimension removal on Qwen1.5-MoE-A2.7B. Values are counts (percentage). Real reasoning collapses by $14\times$ while echo behaviors and empty outputs emerge.

| **Category** | **base model** | **top-12** |
|---|---|---|
| Real reasoning | 791 (60.0%) | 55 ( 4.2%) |
| Partial reasoning | 100 ( 7.6%) | 28 ( 2.1%) |
| Exact echo | 0 ( 0.0%) | 30 ( 2.3%) |
| Truncated echo | 45 ( 3.4%) | 197 (14.9%) |
| Garbled echo | 9 ( 0.7%) | 183 (13.9%) |
| Numeric only | 0 ( 0.0%) | 1 ( 0.1%) |
| Empty output | 0 ( 0.0%) | 99 ( 7.5%) |
| Other | 374 (28.4%) | 726 (55.0%) |
| **Accuracy** | 35.9% | 0.8% |

*(2) Echo behaviors emerge as the dominant failure mode.* Exact, truncated, and garbled echoes together account for 31.1% of outputs at top-12 removal but only 4.1% at top-1. The model defaults to copying or lightly transforming the input rather than generating a response. Garbled echoes typically substitute names while preserving sentence structure (e.g., "Gissela" $\rightarrow$ "Gordy"), suggesting

that surface-level next-token prediction survives while task-conditioned generation does not.

*(3) Empty outputs emerge.* At top-12 removal, 99 outputs (7.5%) contain no tokens, a behavior absent at top-1. Removing the critical dimensions occasionally pushes the next-token distribution into an immediate end-of-sequence state at the very first decoding step.

**Representative examples.** The qualitative gap between the two settings is best illustrated by paired outputs.

*Base model (real reasoning, correct):*

> **Q:** Edgar eats 18 pretzels a day. If his brother eats 1/2 as many, how many does his brother eat in a week?
> **Output:** *Edgar's brother eats 1/2 as many pretzels as Edgar, which means he eats* $18/2 = 9$ *pretzels a day. To find out how many pretzels his brother eats in a week, we multiply* ... $9 \times 7 = 63$ *pretzels in a week. Therefore, his brother eats 63 pretzels in a week.*
> **Pred:** 63 **Gold:** 63 (correct)

*top-12 removal (truncated echo):*

> **Q:** The farm has 30 cows and the zoo has 20 sheep. The zoo has twice as many cows as the farm does ...
> **Output:** *The farm has 30 cows and the zoo has 20 sheep. The zoo has twice as many cows as the farm does* ... (verbatim until truncation)
> **Pred:** 20 **Gold:** 120

*top-12 removal (garbled echo):*

> **Q:** Gissela, Gordy, and Gary are truck drivers. Gissela has a truck large enough to haul 4,000 pounds ...
> **Output:** *Gordy, Gordy, and Gary are truck drivers. Gissela has a truck large enough to haul 4,000 pounds* ...
> **Pred:** 800 **Gold:** 2,800

**Interpretation.** The shift between base model and top-12 is qualitative, not just quantitative: removing $\sim$12 additional intermediate dimensions does not gradually degrade reasoning but eliminates the model's ability to transition from input-conditioned copying to forward generation. The combined 86% share of echo and "other" fragment outputs indicates that the critical dimensions identified by Fisher importance participate in the computation mapping question representations to multi-step reasoning trajectories, rather than in the surface-level token-prediction circuit that produces fluent text. This view is consistent with the cross-domain dissociation in Table 1: multiple-choice knowledge tasks, which rely on input-conditioned token prediction, retain 90–98% of base accuracy, while generation-heavy tasks, which require multi-step trajectories, collapse to 3–13%.

## K Mechanisms of Compression-Induced Improvement: A Unified Fisher-Prior Account

We observe that intermediate dimension compression yields modest accuracy gains on several benchmarks (MultiArith, CMMLU, CEval) despite removing 50% of routed-expert FFN parameters. We attribute these gains to a single underlying mechanism—empirical Fisher importance systematically removes high-prior, low-evidence dimensions while preserving reasoning-supporting dimensions—and demonstrate that the magnitude of improvement on each benchmark is governed by how much *shortcut headroom* the base model's failure modes leave behind. We examine this through output category distributions across benchmarks.

### K.1 Shortcut Suppression on Generation Tasks

On unconstrained reasoning tasks, the base model emits template shortcuts and question paraphrase for 20.2% of MultiArith outputs (Short direct answer 15.5% + Garbled echo 4.7%), against 71.5% Real reasoning. We adopt eight output categories (*real reasoning*: explicit multi-step chain-of-thought reaching a numeric conclusion; *partial reasoning*: reasoning tokens present but incoherent or incomplete; *short direct answer*, *garbled echo*, etc.). Intermediate dimension compression shifts this distribution: *Short direct answer* and *Garbled echo* drop 20.2pp to nearly zero, while *Real reasoning* rises to 98.0%–98.2%. The accompanying accuracy gain (+13.5% to +14.7% from Fisher-IntDim-L to Fisher-IntDim-G) shows that our pruning further disrupts these residual shortcuts even on top of an already-competent base.

This distributional shift is quantitatively reflected in generation lengths: base outputs are bimodal (median 44 tokens, SD 106.6)—a peak of one-line shortcut answers plus a long tail of garbled echoes capped at the 500-token generation limit ($p_{95} = 499$)—while Fisher-IntDim-G outputs are unimodal (median 54, SD 59.0, $p_{95} = 96$), eliminating both extremes. On problem #257 ("Will had \$83, spent \$47; how many \$4 toys can he buy?"), the base model emits only "10" with no derivation; Fisher-IntDim-G writes three explicit steps ($83 - 47 = 36$; $36/4 = 9$) and answers correctly. We attribute this to a *Fisher–prior asymmetry*: shortcut behavior reflects high-prior, low-evidence outputs whose gradients w.r.t. expert FFN parameters are small, while multi-step reasoning requires precise intermediate-state propagation and produces large gradients. Empirical Fisher therefore systematically retains reasoning-supporting dimensions and discards prior-driven shortcut dimensions, exposing latent reasoning capacity that the base model fails to invoke.

Table 27: Detailed Prediction Analysis on MultiArith

| Category | Base | Fisher-IntDim-L | Fisher-IntDim-G |
|---|---|---|---|
| Real reasoning | 429 (71.5%) | 588 (98.0%) | 589 (98.2%) |
| Partial reasoning | 37 ( 6.2%) | 12 ( 2.0%) | 10 ( 1.7%) |
| Short direct answer | 93 (15.5%) | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 28 ( 4.7%) | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 13 ( 2.2%) | 0 ( 0.0%) | 1 ( 0.2%) |
| **Accuracy** | **76.67%** | 90.17% | 91.33% |

### K.2 Format Compliance on Multiple-Choice Tasks

The same Fisher–prior mechanism produces a second manifestation on multiple-choice benchmarks, where the base model's high-prior failure mode is to echo the option text instead of emitting a single answer letter. On CMMLU, 23.6% of base outputs are option-text echoes; on CEval, 25.6%. Both are high-prior, low-evidence emissions: the model copies salient input spans rather than committing to a letter. Fisher-IntDim suppresses these echoes (CMMLU: 23.6% $\rightarrow$ 5.9%, $-17.7$pp; CEval: 25.6% $\rightarrow$ 10.5%, $-15.1$pp) and shifts probability mass to the single-letter format, yielding accuracy gains of +1.3–4.2pp on CMMLU and +2.0pp on CEval. The smaller magnitude relative to MultiArith reflects the smaller fraction of recoverable mass: for option-text echoes, the underlying answer choice is often already wrong, so format correction alone cannot rescue accuracy.

Table 28: Detailed Prediction Analysis on CMMLU

| Category | Base | Fisher-IntDim-L | Fisher-IntDim-G |
|---|---|---|---|
| Short direct answer | 8769 (75.7%) | 10869 (93.8%) | 10862 (93.8%) |
| Truncated echo (text) | 2738 (23.6%) | 698 ( 6.0%) | 678 ( 5.9%) |
| Other | 45 ( 0.4%) | 9 ( 0.1%) | 10 ( 0.1%) |
| **Accuracy** | 60.74% | 62.00% | 64.94% |

Table 29: Detailed Prediction Analysis on CEval

| Category | Base | Global |
|---|---|---|
| Short direct answer | 9048 (73.3%) | 10996 (89.1%) |
| Truncated echo (text) | 3155 (25.6%) | 1296 (10.5%) |
| **Accuracy** | 59.30% | 61.32% |

### K.3 Partial-Reasoning Completion on Long-CoT Benchmarks

The shortcut-suppression account above applies to base models that frequently default to terse or template-style outputs. A natural question is whether the same Fisher–prior mechanism produces analogous effects on frontier-scale reasoning models, where the dominant failure mode is not shortcut emission but *incomplete chain-of-thought*: the model initiates a multi-step reasoning trace but fails to reach a coherent numeric conclusion.

We examine this through output category distributions on AIME 2024, AIME 2025, and Olympiad Bench for Qwen3-30B-A3B and Qwen3.5-35B-A3B, evaluated with avg@8 sampling (240 samples per benchmark, 30 problems $\times$ 8 draws).

**Dominant failure mode shifts from shortcut to partial reasoning.** Unlike the small base models in §K.1, neither Qwen3-30B-A3B nor Qwen3.5-35B-A3B produces echo, numeric-only, or empty outputs at any measurable rate. The sole failure mode is *partial reasoning*: chains that contain valid reasoning tokens but are incomplete or internally incoherent. This accounts for 24.6% and 35.4% of Qwen3-30B-A3B outputs on AIME 2025 and AIME 2026 respectively, and 7.9% and 5.4% for the stronger Qwen3.5-35B-A3B.

**Compression converts partial reasoning into real reasoning.** Tables 30–34 show that Fisher-IntDim-G shifts partial reasoning to real reasoning across all five settings with no new failure modes introduced. The accuracy gains are consistent with the recoverable-mass account: Qwen3-30B-A3B, which starts from a higher partial-reasoning rate, benefits more (+26.7pp on AIME 2025, +6.7pp on AIME 2026) than Qwen3.5-35B-A3B (+10.0pp and +6.7pp), whose partial-reasoning rate is already low.

**Mechanism: Fisher importance targets incomplete-chain dimensions.** We interpret this under the same Fisher–prior account. In a long-CoT reasoning model, *partial reasoning* represents a regime where intermediate-state propagation partially succeeds—the model initiates a plausible chain—but fails to sustain the precise token-to-token dependencies required to close it. Dimensions that support only the initialization of a reasoning chain without contributing to its completion receive lower Fisher scores: their gradient signal across calibration examples reflects high-prior behavior (starting a chain is likely regardless of the specific problem) rather than low-evidence, problem-conditioned computation (completing it). Removing these dimensions selectively suppresses the partial-chain attractor, exposing the model's latent capacity to produce complete derivations.

Table 30: Distribution of AIME 2025 output categories under base Qwen3-30B-A3B and Fisher-IntDim-G (50% routed-FFN compression). 240 samples (30 problems $\times$ avg@8). Pruned model produces *deeper* multi-step CoT than the base.

| Category | Base model | Fisher-IntDim-G (ours) |
|---|---|---|
| Real reasoning | 180 (75.0%) | **198 (82.5%)** |
| Partial reasoning | 59 (24.6%) | 41 (17.1%) |
| Exact echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Truncated echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 1 ( 0.4%) | 1 ( 0.4%) |
| Numeric only | 0 ( 0.0%) | 0 ( 0.0%) |
| Empty output | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 0 ( 0.0%) | 0 ( 0.0%) |
| **Accuracy** | 33.33% (10/30) | **60.00% (18/30)** |

Table 31: Distribution of AIME 2026 output categories under base Qwen3-30B-A3B and Fisher-IntDim-G. 240 samples (30 problems $\times$ avg@8).

| Category | Base model | Fisher-IntDim-G (ours) |
|---|---|---|
| Real reasoning | 155 (64.6%) | **203 (84.6%)** |
| Partial reasoning | 85 (35.4%) | 36 (15.0%) |
| Exact echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Truncated echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Numeric only | 0 ( 0.0%) | 0 ( 0.0%) |
| Empty output | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 0 ( 0.0%) | 1 ( 0.4%) |
| **Accuracy** | 50.00% (15/30) | **56.67% (17/30)** |

**Contrast with the small-model setting.** The long-CoT case and the small-model case share the same underlying mechanism—Fisher-MoE removes high-prior, low-evidence dimensions—but differ in which prior is suppressed. In small base models the prior is a surface-level output template (echo, one-line answer). In frontier reasoning models the prior is an *incomplete chain initialization*: the model defaults to beginning a plausible-looking reasoning trace without the problem-specific precision to complete it. In both cases the recoverable accuracy gain is determined by how large a fraction of outputs are trapped in the high-prior attractor, and compression releases exactly that fraction. The clean absence of echo or empty-output failures in Tables 30–34 further confirms that the removed dimensions are genuinely redundant: unlike critical-

Table 32: Distribution of AIME 2025 output categories under base Qwen3.5-35B-A3B and Fisher-IntDim-G. 240 samples (30 problems × avg@8).

| Category | Base model | Fisher-IntDim-G (ours) |
|---|---|---|
| Real reasoning | 221 (92.1%) | **237 (98.8%)** |
| Partial reasoning | 19 ( 7.9%) | 2 ( 0.8%) |
| Exact echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Truncated echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 0 ( 0.0%) | 1 ( 0.4%) |
| Numeric only | 0 ( 0.0%) | 0 ( 0.0%) |
| Empty output | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 0 ( 0.0%) | 0 ( 0.0%) |
| **Accuracy** | 66.67% (20/30) | **76.67% (23/30)** |

Table 33: Distribution of AIME 2026 output categories under base Qwen3.5-35B-A3B and Fisher-IntDim-G. 240 samples (30 problems × avg@8).

| Category | Base model | Fisher-IntDim-G (ours) |
|---|---|---|
| Real reasoning | 227 (94.6%) | **239 (99.6%)** |
| Partial reasoning | 13 ( 5.4%) | 1 ( 0.4%) |
| Exact echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Truncated echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Numeric only | 0 ( 0.0%) | 0 ( 0.0%) |
| Empty output | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 0 ( 0.0%) | 0 ( 0.0%) |
| **Accuracy** | 66.67% (20/30) | **73.33% (22/30)** |

dimension removal, pruning the bottom 50% by Fisher score introduces no new failure modes in either model family.

### K.4 Summary: One Mechanism, Three Manifestations

Across all benchmarks examined, intermediate-dimension Fisher pruning suppresses the same class of behavior: high-prior, low-evidence outputs that the model produces when it defaults to surface-level continuation rather than task-conditioned generation. The mechanism manifests differently depending on the model family and benchmark format, but the underlying logic is identical in each case.

- **Unconstrained arithmetic (MultiArith).** The base model's dominant failure mode is one-line shortcut answers and garbled echoes. Compression removes the dimensions that support these high-prior, low-computation outputs, shifting the majority of shortcut outputs to complete multi-step derivations.
- **Multiple-choice knowledge (CMMLU/CEval).** The base model echoes full option text instead of committing to a single answer letter in roughly one quarter of outputs. Compression substantially reduces this echo rate, but the accuracy gain is smallest here because format correction alone rarely rescues semantically wrong answers.
- **Long-CoT math reasoning (AIME/Olympiad).** In frontier-scale reasoning models, echo and shortcut behaviors are absent entirely. The sole failure mode is partial reasoning: chains that initiate plausibly but fail to close. Compression converts partial chains to complete derivations, with gain magnitude proportional to the base model's partial-reasoning rate—the model with a higher baseline partial-reasoning rate benefits more, consistent with the recoverable-mass account.

Table 34: Distribution of Olympiad Bench output categories under base Qwen3.5-35B-A3B and Fisher-IntDim-G. 674 samples (single-sample per problem).

| Category | Base model | Fisher-IntDim-G (ours) |
|---|---|---|
| Real reasoning | 613 (90.9%) | **655 (97.2%)** |
| Partial reasoning | 56 ( 8.3%) | 19 ( 2.8%) |
| Exact echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Truncated echo | 0 ( 0.0%) | 0 ( 0.0%) |
| Garbled echo | 5 ( 0.7%) | 0 ( 0.0%) |
| Numeric only | 0 ( 0.0%) | 0 ( 0.0%) |
| Empty output | 0 ( 0.0%) | 0 ( 0.0%) |
| Other | 0 ( 0.0%) | 0 ( 0.0%) |
| **Accuracy** | 77.89% (525/674) | **79.53% (536/674)** |

The magnitude of accuracy gain in each case is governed by the *recoverable mass*: how large a fraction of outputs are trapped in the high-prior attractor, and how directly format or chain-completion correction translates to accuracy. Table 35 summarizes the three manifestations.

Table 35: Summary of the Fisher–prior mechanism across benchmark types.

| Benchmark | Base failure mode | Suppressed by compression |
|---|---|---|
| MultiArith | Shortcut / garbled echo | Shortcuts nearly eliminated |
| CMMLU / CEval | Option-text echo | Echo rate substantially reduced |
| AIME / Olympiad | Partial reasoning (CoT) | Partial chains converted to complete |

We emphasize that none of these gains reflect added capability. They reflect removal of a generation-time prior that suppresses latent circuits already present in the model. The long-CoT results further strengthen this interpretation: in Tables 30–34, compression introduces zero new failure modes while monotonically shifting outputs toward complete reasoning, confirming that the removed dimensions are genuinely redundant rather than load-bearing. On benchmarks limited by raw reasoning

capacity rather than shortcut headroom (MATH, MMLU, BBH), Fisher-IntDim incurs the expected modest capacity cost (Table 3).

## L Dense LLM Pruning Baseline Analysis

Table 36: Comparison on Qwen1.5-MoE-A2.7B at 50% sparsity (zero-shot, T=0). Wanda (Sun et al., 2024b) and SparseGPT (Frantar and Alistarh, 2023) with our method.

| Method | MMLU | HumanEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| *Qwen1.5-MoE-A2.7B* | | | | | | | | | |
| Base | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 | 38.8 |
| Wanda (2:4) | 55.7 | 23.8 | 20.0 | 58.0 | 59.3 | 6.6 | 35.6 | 30.0 | 36.1 |
| SparseGPT (2:4) | 55.9 | 24.4 | 21.3 | 57.8 | 58.5 | 7.4 | 33.1 | 30.0 | 36.0 |
| **Fisher-IntDim-G (ours)** | 49.0 | 34.2 | 24.1 | 61.3 | 64.9 | 9.8 | 27.2 | 28.2 | 37.3 |

Wanda (Sun et al., 2024b) and SparseGPT (Frantar and Alistarh, 2023) are post-training weight compression methods originally designed for dense decoder-only LLMs: Wanda prunes weights by the product of magnitude and input activation norm without weight updates, while SparseGPT formulates compression as a layer-wise sparse regression with second-order information from an approximated Hessian. Both target unstructured or N:M semi-structured patterns (e.g., 2:4) over FFN and attention weight matrices, and are unaware of MoE-specific structural properties such as routing dynamics or expert grouping—when applied to MoE, they degenerate into per-row compression of individual expert FFN matrices, ignoring inter-expert redundancy. Consequently, Lu et al. (2024b) report that directly applying Wanda 2:4 to Mixtral 8x7B causes substantial drops on general LM Harness benchmarks and a near-collapse on math reasoning, suggesting that weight-level semi-structured sparsity inadequately preserves task-specific expert specialization. In addition, Wanda and SparseGPT rely on sparse GEMM, which typically provides limited speedup for MoE models. In contrast, Fisher-MoE preserves dense GEMM execution and achieves more consistent and substantial speedups, as shown in § 3.3.

Our results on Qwen1.5-MoE-A2.7B (Table 36) exhibit the same trend: both Wanda and SparseGPT retain moderate average performance but suffer pronounced degradation on math and code generation, while Fisher-IntDim-G (ours) achieves comparable averages with stronger preservation on knowledge-heavy and code benchmarks. This complementarity reflects a fundamental design difference—weight-level 2:4 sparsity preserves all experts while sparsifying their internal weights, whereas our method exploits MoE-aware intermediate dimension structure to selectively retain task-critical intermediate dimensions across experts.

## M Model Size and Whole-Model Compression

In this section, we report the total parameter counts and disk footprints of several MoE backbones before and after applying intermediate dimension compression at a target MoE compression ratio of $p = 50\%$.

It is important to distinguish between the *MoE compression ratio* $p$ and the resulting *whole-model compression rate* $p_{\text{model}}$. While $p$ refers to the fraction of parameters removed within the expert FFN modules, $p_{\text{model}}$ measures the reduction in total model parameters. The latter is consistently smaller because several components are not compressed, including attention layers, the shared expert, embeddings, routing networks, and layer normalization parameters.

As shown in Table 37, intermediate dimension compression achieves substantial reductions in total model size across all backbones, with whole-model compression rates ranging from approximately $43\%$ to $48\%$. Notably, these reductions translate directly into proportional decreases in disk storage requirements, making the compressed models significantly more efficient for deployment without modifying the model architecture or routing structure.

Table 37: Parameters and disk size for several models before and after compression at MoE compression ratio $p = 50\%$. The *whole-model compression rate* $p_{\text{model}}$ is the percentage reduction in total parameter count; it is smaller than $p$ and varies across backbones because attention, the shared expert, embeddings, the router, and layer norms are not compressed.

| Model | Parameters | Disk (GB) | Whole-model rate $p_{\text{model}}$ |
|---|---|---|---|
| Qwen1.5-MoE-A2.7B | 14.3B | 28.6 | 43.5% |
| Qwen1.5-MoE-A2.7B (compressed) | 8.1B | 16.2 | |
| OLMoE-1B-7B-0125 | 6.9B | 13.8 | 46.6% |
| OLMoE-1B-7B-0125 (compressed) | 3.7B | 7.4 | |
| Qwen3-30B-A3B | 30.5B | 61.1 | 47.5% |
| Qwen3-30B-A3B (compressed) | 16.0B | 32.1 | |
| Qwen3.5-35B-A3B | 36.0B | 71.9 | 48.4% |
| Qwen3.5-35B-A3B (compressed) | 18.6B | 37.1 | |

## N Divergence of Expert Selection Across Importance Metrics

To better understand how different importance metrics affect expert-level pruning decisions, we analyze the overlap between the sets of retained experts selected by three methods: Fisher-based importance (A), activation-based importance (B), and score-based importance (C).

Let $\mathcal{E}_\ell^{(m)}$ denote the set of retained experts at layer $\ell$ under method $m$. We quantify similarity using two complementary measures: (i) the Jaccard similarity over the union of retained experts across all layers, and (ii) the average per-layer overlap, defined as the number of shared experts per layer.

Table 38 summarizes the results.

**Fisher selects substantially different experts.** The overlap between Fisher-based pruning and both activation- and score-based methods is low, with Jaccard similarities of only $28.5\%$ and $29.1\%$, respectively. At the layer level, Fisher shares on average only $\sim 13$ out of 30 experts with these methods. In contrast, activation- and score-based pruning exhibit much higher agreement (Jaccard $54.9\%$, $\sim 21.5/30$ overlap), indicating that these heuristic metrics tend to select similar experts.

**Limited consensus across all methods.** The intersection across all three methods contains only $8.8$ experts per layer on average, representing less than one-third of the expert pool. This further highlights the diversity of expert importance signals captured by different metrics.

**Implications for compression.** These results suggest that Fisher-based importance captures a fundamentally different notion of expert utility compared to activation- or score-based heuristics. In particular, the low overlap indicates that Fisher is not merely a refinement of existing metrics, but rather identifies distinct experts that may be critical for preserving performance. This observation provides insight into why Fisher-guided compression outperforms prior expert-level pruning approaches.

Table 38: Jaccard similarity and average per-layer overlap of removed expert sets across different pruning methods.

| Comparison | Jaccard | Avg. Overlap / Layer |
|---|---|---|
| A∩B (Fisher vs Activation, same GSM) | 28.5% | 13.3/30 |
| A∩C (Fisher vs Score) | 29.1% | 13.7/30 |
| B∩C (Activation vs Score) | 54.9% | 21.5/30 |
| A∩B∩C (All three retained) | N/A | 8.8/30 |

## O Comparison Against the Dense Baseline and Uncompressed MoE

To place Fisher-MoE in the broader context of dense-vs-sparse model trade-offs, we compare three models of comparable total parameter budget: (i) **Qwen1.5-7B**, a dense transformer; (ii) **Qwen1.5-MoE-A2.7B**, the uncompressed sparse MoE that serves as our base model; and (iii) **Fisher-MoE**, our compressed model derived from Qwen1.5-MoE-A2.7B by removing 50% of every routed expert's FFN intermediate dimensions using Fisher-IntDim-E with 128 GSM8K calibration samples (`moe_intermediate_size`: $1408 \rightarrow 704$). We report task accuracy, activated/total parameters, and standalone throughput on a single NVIDIA A100-80G following the same setting as Qwen official blogpost (Qwen Team, 2024).

**Setup.** All accuracies use vLLM v0.8.4 in bfloat16 with `max_model_len=` 4096, `temperature=` 0.1, and `seed=` 1234. MBPP uses `-stop_sequences "[DONE], END"`. The throughput benchmark fixes input length to 1,000 tokens and output length to 1,000 tokens on a single A100-80G.

**Parameter and activation budget.** Table 39 compares parameter counts. While Fisher-MoE and Qwen1.5-7B occupy comparable total budgets (8.1 B vs. 7.7 B), Fisher-MoE activates only **2.27 B** parameters per token—about 29% of the dense activation cost—because at most $K{=}4$ of $N{=}60$ routed experts fire per token after intermediate dimension compression.

Table 39: Parameter counts for the dense baseline and the two MoE models. Activated parameters are non-expert parameters plus $K{=}4$ active routed experts per layer. Counts are extracted directly from safetensors headers.

| Metric | Qwen1.5-7B | Qwen1.5-MoE-A2.7B | Fisher-MoE |
|---|---|---|---|
| Total parameters | 7.72 B | 14.32 B | 8.09 B |
| Activated parameters / token | 7.72 B | 2.69 B | **2.27 B** |
| Activated non-embedding params | 6.48 B | 2.07 B | 1.65 B |
| Embedding + LM head | 1.25 B | 0.62 B | 0.62 B |
| Routed experts (total / active) | N/A | 60 / 4 | 60 / 4 |
| MoE intermediate size per expert | N/A | 1,408 | 704 ($-50\%$) |
| Architecture | Dense | Sparse MoE | Sparse MoE (compressed) |

**Task accuracy.** Table 40 reports eight-task accuracy. Fisher-MoE retains the math performance of the dense baseline, it slightly *exceeds* the dense Qwen1.5-7B on GSM8K (35.0 vs. 34.1) and matches it on MATH (8.0 vs. 8.2) while losing modest ground on knowledge-heavy and coding tasks (MMLU $-8.2$, HumanEval $-12.3$). The average drop from dense is roughly 5 points despite activating $3.4\times$ fewer parameters per token.

Table 40: Eight-task accuracy (%) for the dense baseline, the uncompressed Qwen1.5-MoE-A2.7B, and Fisher-MoE compressed at the 50% MoE compression ratio with in-domain calibration. Higher is better. Same vLLM decoding settings as the main experiments. Qwen1.5-MoE-A2.7B numbers are reproduced from the *base* row of Table 3.

| Model | MMLU | HEval | MBPP | CEval | CMMLU | MATH | GSM8K | BBH | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| Qwen1.5-7B (dense) | 58.5 | 33.5 | 21.1 | 67.9 | 69.6 | 8.2 | 34.1 | 29.9 | 40.4 |
| Qwen1.5-MoE-A2.7B (base) | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 | 38.8 |
| Fisher-MoE (50% compression ratio.) | 50.3 | 21.2 | 20.5 | 62.0 | 61.6 | 8.0 | **35.0** | 28.3 | 35.9 |

**Throughput and tokens-per-second.** Table 41 reports vLLM throughput on a single A100-80G with 1,000 input and 1,000 output tokens. Qwen1.5-MoE-A2.7B already runs $\sim 1.74\times$ faster than the dense Qwen1.5-7B because of its sparse activation pattern and shared expert. Fisher-MoE pushes this further: by halving each routed expert's intermediate dimension, the per-token expert FLOPs drop accordingly, yielding $2.10\times$ **the throughput** of the dense baseline and $1.21\times$ **the throughput** of the uncompressed MoE.

Table 41: Standalone vLLM throughput on a single NVIDIA A100-80G with 1,000 input tokens and 1,000 output tokens. Higher is better. "Speedup" columns are relative to Qwen1.5-7B and Qwen1.5-MoE-A2.7B respectively.

| Model | Throughput (req/s) | TPS (tok/s) | vs. 7B | vs. MoE-A2.7B |
|---|---|---|---|---|
| Qwen1.5-7B-Chat | 1.15 | 2,298.89 | $1.00\times$ | $0.57\times$ |
| Qwen1.5-MoE-A2.7B-Chat | 2.01 | 4,010.27 | $1.74\times$ | $1.00\times$ |
| Fisher-MoE | **2.42** | **4,853.52** | **2.10**$\times$ | **1.21**$\times$ |

At a comparable total-parameter budget, Fisher-MoE delivers the strongest inference profile of the three: $2.10\times$ the dense baseline throughput, 29% of its activated-parameter cost, and accuracy competitive with the dense model on the math reasoning tasks (GSM8K, MATH). Compared to the uncompressed Qwen1.5-MoE-A2.7B, removing 50% of routed-expert intermediate dimensions buys an additional $1.21\times$ throughput on top of the MoE baseline's already substantial inference advantage—demonstrating that intermediate-dimension compression is complementary to, rather than a dilution of, the inference benefits of sparse architectures.

## P Theoretical Comparison: Empirical Fisher, Diagonal Hessian, and First-Order Pruning

A reasonable concern about our use of the empirical Fisher as the importance metric is that "Fisher importance" is related to Hessian-based attribution criteria. This appendix consolidates the relationships and reports the controlled comparison that supports this hedged framing.

### P.1 What the Empirical Fisher Computes

With $\mathcal{L}(x, y) = -\log p_\theta(y \mid x)$, the empirical Fisher (Eq. 4) is the diagonal of $\frac{1}{N}\sum_{(x,y)\in\mathcal{D}} \nabla_\theta \mathcal{L}(x, y)\nabla_\theta \mathcal{L}(x, y)^\top$. Two facts are directly relevant.

**(i) Empirical Fisher.** For a per-parameter score, the empirical Fisher reduces to $s^{\text{Fisher}}(\theta_i) = \frac{1}{N}\sum_{(x,y)}(\partial\mathcal{L}/\partial\theta_i)^2$. This is the Taylor-expansion saliency: at a parameter we expect $\Delta\mathcal{L} \approx |\partial\mathcal{L}/\partial\theta_i| \cdot |\delta_i|$, so empirical fisher is the variance of this first-order increment over the calibration distribution. In our setting the two metrics coincide up to a constant after we group parameters into intermediate dimension units (Eq. 5), and the practical advantage we report over magnitude/activation/router heuristics is therefore equally a statement about first-order pruning under the same intermediate dimension grouping.

**(ii) Empirical Fisher vs. true Fisher vs. diagonal Hessian.** The Fisher information matrix is $F_\theta = \mathbb{E}_x\mathbb{E}_{y\sim p_\theta}[\nabla_\theta \log p_\theta \, \nabla_\theta \log p_\theta^\top] = -\mathbb{E}_x\mathbb{E}_{y\sim p_\theta}[\nabla^2_\theta \log p_\theta(y \mid x)]$, so at the model's optimum on its own data distribution, the (true) Fisher and the negative Hessian of the log-likelihood coincide. The *empirical* Fisher samples $y$ from the data rather than from the model, which makes it a biased estimate of either quantity away from the optimum (Kunstner et al., 2019); for a frozen, pretrained MoE evaluated on a calibration set the bias can be non-negligible.

### P.2 Relationship to Hessian-Based Pruning (OBD/OBS, SparseGPT/Wanda)

Optimal Brain Damage (OBD, LeCun et al. (1989)) and Optimal Brain Surgeon (OBS, Hassibi and Stork (1992)) score weight $\theta_i$ by $s_i^{\text{OBD}} = \frac{1}{2}H_{ii}\,\theta_i^2$, the second-order Taylor estimate of the loss increase under a single-weight perturbation. Modern weight-pruning methods such as SparseGPT (Frantar and Alistarh, 2023) and Wanda (Sun et al., 2024b) replace the parameter Hessian with a layer-wise input second-moment matrix, but the spirit is the same: rank parameters by curvature×magnitude. Two practical differences with our criterion:

- **Curvature vs. sensitivity.** OBD/OBS rank parameters by the second-order loss change after the optimal compensating update. Empirical Fisher rank parameters by the first-order loss change with no compensation. The two coincide at an optimum and on the data distribution the model was trained on, but disagree off-optimum and under domain mismatch.
- **Aggregation unit.** OBD/OBS and SparseGPT/Wanda are derived for unstructured pruning of single weights or layer-input groupings. Our contribution (Eq. 5) groups parameters tied to a single FFN intermediate dimension across the gate/up/down slabs of a routed expert. This grouping is what enables structurally smaller MoE inference (§3); it is orthogonal to the choice between Fisher- and Hessian-based scoring rules and could be paired with either.

WandB and SparseGPT are primarily designed for dense model pruning and sparsification, and thus fall outside the scope of MoE compression. Nevertheless, we include comparisons with these methods in Appendix L to demonstrate the effectiveness of the proposed Fisher-MoE.

### P.3 Empirical Comparison Under the Same Grouping

To check that the gains we report are not simply a relabeling of an existing scoring rule, we compare three metrics on Qwen1.5-MoE-A2.7B at $p = 50\%$ under *exactly* the intermediate dimension grouping of Eq. 5, with the same calibration data (GSM8K training set, 128 samples), the same compression operation, and the same evaluation protocol. Only the per-parameter score differs:

- **Magnitude** (data-free): $|\theta_i|$, the baseline signal.
- **First-order $|\nabla\mathcal{L}|$ (no square)**: $\frac{1}{N}\sum|\partial\mathcal{L}/\partial\theta_i|$, classical first-order pruning without squaring.
- **Empirical Fisher / squared gradient (ours)**: $\frac{1}{N}\sum(\partial\mathcal{L}/\partial\theta_i)^2$ (Eq. 4).

Computing a faithful diagonal-Hessian variant on a 14B-parameter MoE is several times the cost of empirical Fisher (a Hutchinson estimator requires multiple Hessian–vector products per calibration sample); we discuss the relationship theoretically above and treat a full empirical sweep as future work.

Empirical Fisher is best understood as a coordinate-wise squared-gradient score under the intermediate dimension grouping that is our actual contribution. Our contribution lies in the attribution unit and grouping (Eq. 5) and the structural-removal operation it enables (§3), not in the choice of scoring rule against the broader family of gradient- and Hessian-based criteria.

## Q Where is the Redundancy? Locating Compressible Substructures in an MoE

The intermediate dimension granularity is one of three structural axes along which a sparse MoE could plausibly be compressed. To justify our choice empirically, we run a controlled head-to-head comparison: we hold the importance signal fixed (**Fisher importance**, §2.1), the backbone fixed (**Qwen1.5-MoE-A2.7B**), and the calibration data fixed (domain-matched, 128 samples), and we vary only *which structural unit Fisher importance is applied to*. We consider three axes:

1. **Expert-level pruning** – Fisher scores are aggregated per routed expert, and the lowest-ranked experts are removed wholesale. We further study the gate/up/down sub-blocks individually (*Fisher-UP*, *Fisher-DOWN*, *Fisher-GATE*, *Fisher-UP+GATE*) and a router-side variant (*MoE compression (Fisher)*) to localize the contribution within an expert.

2. **Intermediate dimension pruning** – Fisher scores are aggregated per FFN intermediate dimension across the gate/up/down slabs of every routed expert (Fisher-IntDim, our **Fisher-MoE** method).

3. **Attention-head pruning** – Fisher scores are aggregated per attention head, and the lowest-ranked heads are removed.

For each axis we sweep a fixed compression budget and report eight downstream tasks. To put expert-level pruning on the strongest possible footing, we also include the activation-, score-, and magnitude-based heuristics from prior work as additional expert-level baselines, and an alternative *router DenseMixer* variant. Table 42 reports the full sweep.

**Expert-level pruning is uniformly fragile.** Every method that removes whole experts collapses on generation-heavy tasks: GSM8K stays below 23% of base, HumanEval below 50%, and MATH near zero, regardless of whether the importance signal is activation, score, magnitude, Fisher, router-side Fisher, or expert-level Fisher. Decomposing Expert-Fisher further into its three projection slabs (UP, DOWN, GATE) does not help—no slab dominates the others. We read this as evidence that redundancy at expert granularity is *limited*: the units identified for removal are not truly redundant, just less salient on average, so discarding them takes essential computation with them.

**Attention-head pruning is even more fragile.** At 25% head removal, MMLU drops to 43.12% ($\sim$73% of base) and code/math collapse below 13 points. At 50% head removal, the model essentially stops working ($\leq$4% on every task). Attention heads concentrate too much per-head capacity to be removed at this scale without retraining; head-level redundancy in this MoE is effectively zero at the budgets we consider.

**Intermediate dimension pruning has substantial slack.** The contrast with FFN intermediate dimensions is striking. At a *25%* budget, Fisher-IntDim already *exceeds* the uncompressed base on CEval (+10.4), CMMLU (+6.8), and GSM8K (+6.5), and matches base on MMLU/MBPP/BBH within 1–4 points. At the much more aggressive *50%* budget, intermediate dimension pruning still retains 84–116% of base on every non-MATH task, while every expert-level competitor at a comparable MoE compression ratio retains $\leq$25% of base on most generation tasks.

**Conclusion.** Across the same backbone, the same Fisher signal, and the same calibration set, FFN intermediate dimensions are the structural unit with the most slack: 25–50% of them can be removed without meaningful performance loss, whereas pruning the same fraction of experts or attention heads is catastrophic. This empirically grounds the granularity choice in our main paper.

Holding the Fisher signal and calibration fixed, FFN intermediate dimensions admit 25–50% removal at near-zero cost on most tasks, while re-

Table 42: Where is the redundancy? Fisher-guided compression applied at three different structural granularities on Qwen1.5-MoE-A2.7B (domain-matched calibration). Heuristic expert-level baselines (activation/score/magnitude) are included for reference. Higher is better.

| **Method** | **MMLU** | **HumanEval** | **MBPP** | **CEval** | **CMMLU** | **MATH** | **GSM8K** | **BBH** |
|---|---|---|---|---|---|---|---|---|
| base | 59.3 | 32.6 | 24.6 | 59.3 | 60.7 | 13.0 | 35.9 | 30.7 |
| *Expert-level pruning (50% experts removed)* | | | | | | | | |
| MoE Compression (activation) | 26.3 | 0.0 | 0.0 | 42.5 | 43.1 | 0.0 | 1.9 | 24.5 |
| MoE Compression | 37.2 | 0.0 | 0.8 | 13.5 | 32.3 | 0.0 | 1.9 | 23.5 |
| MoE-Pruner | 18.6 | 0.8 | 0.0 | 33.3 | 35.8 | 0.1 | 1.7 | 13.2 |
| MoE compression (Fisher, router-side) | 39.4 | 12.9 | 8.1 | 48.8 | 47.7 | 2.0 | 16.9 | 25.7 |
| Expert-Fisher | 37.9 | 12.9 | 11.7 | 48.6 | 44.2 | 2.4 | 18.0 | 24.1 |
| → Fisher-UP | 39.1 | 9.9 | 12.2 | 47.0 | 44.6 | 2.9 | 22.1 | 19.1 |
| → Fisher-DOWN | 38.7 | 11.4 | 6.7 | 49.0 | 45.1 | 4.1 | 20.1 | 24.0 |
| → Fisher-GATE | 37.4 | 15.9 | 2.2 | 47.3 | 39.5 | 2.6 | 22.3 | 22.5 |
| → Fisher-UP+GATE | 37.5 | 7.6 | 7.8 | 46.6 | 40.8 | 2.8 | 19.3 | 18.2 |
| *FFN intermediate dimension pruning (ours)* | | | | | | | | |
| Fisher-IntDim-E-25% | 55.9 | 30.5 | 23.8 | **69.7** | **67.5** | 11.2 | **36.6** | 29.7 |
| **Fisher-IntDim-E-50%** | 50.3 | 34.2 | 20.5 | 62.0 | 61.6 | 9.8 | 35.0 | 28.2 |
| *Attention-head pruning* | | | | | | | | |
| Attention Head Pruning-25% drop | 43.12 | 12.20 | 1.11 | 35.98 | 34.86 | 0.56 | 2.73 | 3.21 |
| Attention Head Pruning-50% drop | 2.11 | 0.00 | 0.00 | 0.28 | 4.28 | 0.00 | 0.00 | 0.00 |

moving the same fraction of experts or attention heads collapses generation-heavy benchmarks—identifying intermediate dimensions as the locus of structural redundancy in the MoE (Appendix Q).