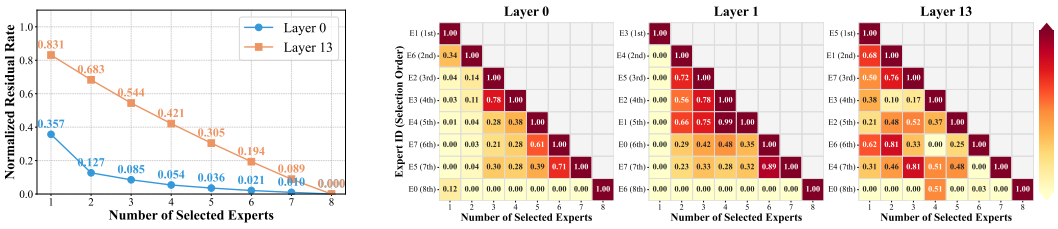
OMP-MoE: Efficient Expert Pruning for Mixture-of-Experts LLMs via Orthogonal Matching Pursuit

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) models enable efficient scaling of large language models but face critical deployment challenges due to massive memory requirements. Existing pruning methods either incur prohibitive search costs or neglect the dynamic interdependencies between experts. To address these challenges, we present OMP-MoE, a novel training-free compression framework for reducing expert redundancy in MoE-based LLMs. Based on observations of expert contribution patterns, we reformulate the pruning problem as a sparse signal reconstruction task solved through Orthogonal Matching Pursuit. Specifically, our method first treats individual expert contributions as dictionary atoms and selects experts that greedily minimize reconstruction error with linear computational complexity. Then, we optimize cross-layer expert allocation through a water-filling strategy that accounts for both reconstruction quality and routing stability. Finally, we introduce OMP-MoE†, an adaptive inference mechanism that dynamically adjusts expert activation based on energy prediction. Comprehensive experiments on Qwen, DeepSeek-V2, GPT-OSS, and Mixtral MoE demonstrate consistent improvements over existing methods at 25-50% pruning ratios. For Qwen3-30B-A3B at 50% expert pruning, we retain 93.3% of original performance, achieving 33$\times$ faster search and 1.55$\times$ inference speedup. Codes are available in Supplementary Material.



(a) Normalized residual curves

(b) Reconstruction gain heatmaps generated by OMP

Figure 1: **Analysis of Dynamic Expert Selection.** We visualize the selection process across three representative layers in Mixtral-8$\times$7B: (**a**) Decay of normalized residual rate curve as a function of the number of selected experts. (**b**) Dynamic importance of candidate experts sorted by selection order. We provide more details and visualizations across various MoE LLMs in Appendix E.1 and E.2.

1 Introduction

The scaling of Large Language Models (LLMs) has led to unprecedented capabilities across diverse tasks (Achiam et al., 2023; Grattafiori et al., 2024). To balance massive parameter counts with computational efficiency, the Mixture-of-Experts (MoE) architecture has emerged as a dominant paradigm Shazeer et al. (2017); Fedus et al. (2022). Modern MoE models, such as Mixtral Jiang et al. (2024), DeepSeek-V3 DeepSeek-AI (2024), Qwen-MoE Yang et al. (2025), and GPT-OSS Agarwal et al. (2025), utilize sparse gating to activate only a subset of experts per token, and build upon

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

foundational works that introduced sparsity to scale capacity Shazeer et al. (2017); Lepikhin et al. (2021); Fedus et al. (2022). However, the immense scale of these models, which exceeds hundreds of billions of parameters, presents significant challenges for deployment in environments with limited resources Ghorbani et al. (2022); Clark et al. (2022).

Existing works on MoE compression primarily follow two trajectories: expert merging and expert pruning. Merge-based methods, such as MC-SMoE Li et al. (2024) and HC-SMoE Chen et al. (2025), attempt to combine redundant experts into a reduced set of "centroid" experts. While Sub-MoE Li et al. (2025d) improves upon this by merging experts within a subspace, recent studies like REAP Lasby et al. (2026) argue that merging inevitably introduces lossy representations by blending distinct specialized features. In contrast, expert pruning directly removes less critical experts to reduce the model's footprint. However, current pruning techniques face a significant efficiency-performance trade-off. Combination-based methods like NAEE Lu et al. (2024) require exhaustive evaluations over combinatorial spaces. For a model with $N_e = 128$ experts and a target of $n = 64$, the search space reaches $\binom{128}{64} \approx 2.4 \times 10^{37}$ combinations, making global optimization prohibitive. Other approaches like HEAPr Li et al. (2025a) leverage Hessian-based second-order information to guide pruning, but calculating the Hessian inverse for massive models incurs prohibitive memory and computational overhead. Similarly, differentiable methods such as DiEP Bai et al. (2025) necessitate additional fine-tuning, which imposes a heavy training burden. These challenges present the key question: *How can we design efficient pruning frameworks that simultaneously achieve fast search and maintain performance?*

"The shortest distance between two points is a straight line."

— Euclid

As the quote suggests, efficient solutions often emerge from direct formulations that exploit underlying structure. Motivated by sparse approximation, we analyze whether expert pruning can be posed as reconstructing the MoE layer output from a small subset of expert contribution atoms. Figure 1 shows two useful properties. First, residual curves decay at different rates across layers, indicating that a uniform expert budget is suboptimal. Second, expert importance changes after each selected expert because the residual signal changes, making static one-shot ranking insufficient. These findings underscore that employing Orthogonal Matching Pursuit (OMP) Pati et al. (1993); Tropp & Gilbert (2007) for expert selection is a promising approach for MoE compression that balances efficiency, optimality, and performance preservation.

Building on these findings, we propose OMP-MoE, a training-free expert pruning framework for MoE LLMs. OMP-MoE formulates expert pruning as a sparse signal reconstruction problem, where expert contributions are treated as dictionary atoms and selected by greedily reducing the current reconstruction residual. This design avoids exhaustive subset search, reduces the search complexity from exponential to linear, and provides a locally optimal greedy selection rule under the binary expert-retention constraint. To handle layer heterogeneity, OMP-MoE further allocates expert budgets across layers with a water-filling strategy that combines reconstruction error and routing stability. We also introduce OMP-MoE†, an adaptive inference mechanism that skips low-energy expert executions for input-aware computation reduction.

Extensive experiments demonstrate that the OMP-MoE framework identifies superior expert subsets and consistently outperforms state-of-the-art methods across multiple architectures. Our experiments reveal three primary advantages. First, OMP-MoE achieves superior performance retention: at 50% pruning on Qwen3-30B-A3B, it maintains 93.3% of the original model's accuracy, substantially outperforming the 53.0% retained by MC-SMoE and the 80.5% retained by NAEE. Second, our method exhibits remarkable search efficiency, completing expert selection in just 641 seconds, a 33$\times$ speedup over NAEE while maintaining comparable memory usage. Third, our adaptive OMP-MoE† delivers tangible inference benefits, achieving 1.46$\times$ end-to-end acceleration on DeepSeek-V2-Lite at 50% pruning. Beyond standard benchmarks, OMP-MoE demonstrates strong capability retention on reasoning-intensive benchmarks, preserving 59.6% performance on GSM8K compared to 26.5% for the best baseline. Furthermore, OMP-MoE exhibits orthogonal compatibility with quantization and unstructured pruning methods.

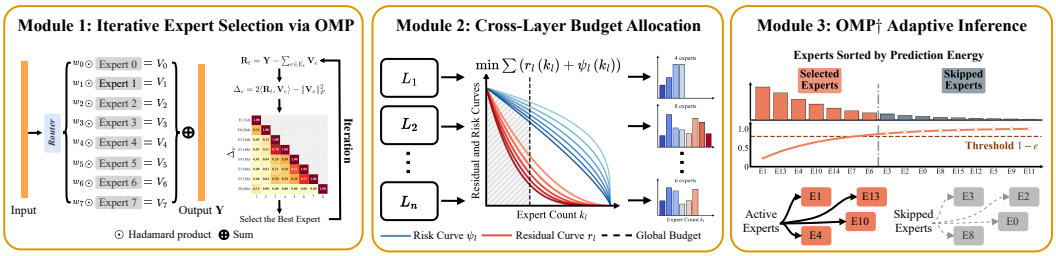


Figure 2: Overview of our OMP-MoE framework: (a) Greedy pursuit of expert contribution atoms through iterative selection to minimize layer wise reconstruction error. (b) Cross-layer resource optimization via water-filling to align expert counts with marginal benefits. (c) Energy-based dynamic pruning during runtime to achieve streamlined inference without performance degradation.

2 Related Work

Mixture-of-Experts Compression. The compression of MoE models has seen rapid development as parameter counts scale. Expert merging methods consolidate weights to reduce memory footprints. MC-SMoE Li et al. (2024) groups and merges experts based on routing patterns, though this fusion often degrades the fine-grained specialization of the original model. HC-SMoE Chen et al. (2025) utilizes hierarchical clustering to optimize similarity metrics, yet its static nature fails to account for dynamic expert interactions during inference. Sub-MoE Li et al. (2025d) employs subspace expert merging to preserve dominant components, but blending distinct expert functionalities remains fundamentally lossy. Expert pruning directly removes redundant parameters to improve efficiency. NAEE Lu et al. (2024) identifies critical experts through combinatorial subset evaluation, which becomes computationally prohibitive as the number of experts increases. DiEP Bai et al. (2025) utilizes differentiable masks for expert selection, necessitating a resource-intensive fine-tuning stage. Shapley-MoE Huang et al. (2025) applies game-theoretic frameworks to estimate marginal contributions, but the exact calculation is often intractable for massive MoE architectures. HEAPr Li et al. (2025a) guides pruning via Hessian-based second-order information, which incurs significant memory and computational overhead for high-dimensional models. In contrast to prior approaches hindered by prohibitive optimization costs or the neglect of inter-expert correlations, OMP-MoE employs a training-free, iterative mechanism to select experts with linear computational complexity.

Orthogonal Matching Pursuit. Orthogonal Matching Pursuit is a classic greedy algorithm designed for sparse signal reconstruction in compressed sensing Tropp & Gilbert (2007); Pati et al. (1993). It iteratively selects dictionary atoms most correlated with the current residual and updates the residual via orthogonal projection. In neural network optimization, OMP has been used for filter pruning in CNNs Yang et al. (2009) and efficient sparse coding for KV caches Kim et al. (2025). We first explore OMP for MoE compression, treating individual experts as dictionary atoms to solve the pruning problem.

3 Methodology

We propose OMP-MoE, a framework designed to efficiently identify the optimal expert subset under the iterative principles of Orthogonal Matching Pursuit. As illustrated in Figure 2, OMP-MoE is structured into three main components: (1) single-layer expert selection via OMP, (2) cross-layer budget allocation with routing stability, and (3) adaptive inference with OMP-MoE†. We provide the detailed theoretical derivation of the OMP-MoE framework and the formal proofs of optimality in Appendix A.

3.1 Preliminary

Mixture-of-Experts Mechanism. Given an MoE layer consisting of N_e experts $\{F_e\}_{e=1}^{N_e}$, the output $\mathbf{Y}$ for an input $\mathbf{X}$ is computed as the sparse weighted sum of expert outputs:

$$\mathbf{Y} = \sum_{e=1}^{N_e} G(\mathbf{X})_e \odot F_e(\mathbf{X}) = \sum_{e=1}^{N_e} \mathbf{V}_e, \tag{1}$$

where $\odot$ denotes the broadcasting element-wise product and $\mathbf{V}_e$ represents the *contribution atom* of expert e. The gating scores $G(\mathbf{X})$ are computed from the routing matrix $\mathbf{W}_g \in \mathbb{R}^{d \times N_e}$ by the architecture-native router:

$$G(\mathbf{X}) = \text{Router}(\mathbf{X}\mathbf{W}_g), \tag{2}$$

where $\text{Router}(\cdot)$ denotes the model's native composition of score transformation, sparse expert selection, and any normalization or bias correction. A common instance applies softmax scores followed by top-k selection, as in Mixtral's top-2 router Jiang et al. (2024). Other architectures such as DeepSeek-V3 DeepSeek-AI (2024) use sigmoid scores and top-8 selection.

In implementation, $\mathbf{X}$ denotes a mini-batch of token representations at a given MoE layer. For B sequences with length T, we flatten them into $N = B \times T$ tokens, so $\mathbf{Y} \in \mathbb{R}^{N \times d}$ and $\mathbf{V}_e \in \mathbb{R}^{N \times d}$ store token-wise layer outputs and weighted expert contributions.

Principles of Orthogonal Matching Pursuit. Orthogonal Matching Pursuit (OMP) Pati et al. (1993) is a classic greedy algorithm designed for sparse approximation:

$$\min_{\mathbf{c}} \|\mathbf{Y} - \mathbf{D}\mathbf{c}\|_F^2 \quad \text{subject to} \quad \|\mathbf{c}\|_0 \leq n, \tag{3}$$

where $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{N_e}]$ is a dictionary of atoms. OMP iteratively selects the atom $\mathbf{d}_i$ that maximizes the projection onto the current residual signal and updates the coefficient vector $\mathbf{c}$ via orthogonal projection to minimize the reconstruction error.

We adopt the OMP-MoE framework by treating the set of contribution atoms $\{\mathbf{V}_1, \ldots, \mathbf{V}_{N_e}\}$ as our dictionary. Crucially, we constrain the optimization coefficients to be binary, i.e., $c_e \in \{0, 1\}$, indicating whether an expert is retained. This yields the following reconstruction objective for expert selection:

$$\min_{E:|E|=n} \left\| \mathbf{Y} - \sum_{e \in E} \mathbf{V}_e \right\|_F^2, \tag{4}$$

where E represents the support set containing the indices of n selected experts. By reformulating pruning as a signal reconstruction task, we transform a combinatorial search into an efficient greedy pursuit of the optimal expert subset.

3.2 Iterative Expert Selection via OMP

To solve the sparse approximation problem in Eq. 4, we adopt the greedy selection strategy from OMP theory. Instead of performing an exhaustive combinatorial search over $C(N_e, n)$ directly, OMP iteratively selects the expert that provides the maximum reduction in the current residual rate.

Algorithm 1 Greedy Expert Selection via OMP

Require: Layer output $\boldsymbol{Y}$, contribution atoms $\{\boldsymbol{V}_e\}_{e=1}^{N_e}$, expert norms $\{\|\boldsymbol{V}_e\|_F^2\}_{e=1}^{N_e}$, target count n
Ensure: Set of retained experts E^{keep}
  Initialize residual $\boldsymbol{R}_0 \leftarrow \boldsymbol{Y}$
  Initialize selected set $E_0 \leftarrow \varnothing$
  for $t = 0$ to $n - 1$ **do**
    $\Delta_e \leftarrow 2\langle \boldsymbol{R}_t, \boldsymbol{V}_e \rangle - \|\boldsymbol{V}_e\|_F^2$ for all $e \notin E_t$
    $e^* \leftarrow \arg\max_{e \notin E_t} \Delta_e$
    $\boldsymbol{R}_{t+1} \leftarrow \boldsymbol{R}_t - \boldsymbol{V}_{e^*}$
    $E_{t+1} \leftarrow E_t \cup \{e^*\}$
  end for
  Return E_n

Analytical Selection Criterion. Let E_t be the set of experts selected after t rounds, and $\mathbf{R}_t = \mathbf{Y} - \sum_{e \in E_t} \mathbf{V}_e$ be the current residual signal (with $\mathbf{R}_0 = \mathbf{Y}$). When evaluating a candidate expert $e \notin E_t$, the reconstruction gain Δ_e can be derived analytically in a single step:

$$\Delta_e = 2\langle \mathbf{R}_t, \mathbf{V}_e \rangle - \|\mathbf{V}_e\|_F^2 \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product. We provide the detailed mathematical derivation of this formula in Appendix A. This criterion is locally optimal under the binary expert-retention

constraint because, conditioned on the current selected set, maximizing Δ_e is exactly equivalent to minimizing the next residual norm. The full proof is given in Appendix A.2.

To identify the critical experts, we employ an iterative greedy search. This process consists of two phases: a one-time *Statistics Phase* that performs a single forward pass to cache the signal $\mathbf{Y}$ and expert contribution atoms $\{\mathbf{V}_e\}_{e=1}^{N_e}$ for each layer, and a *Greedy Selection Phase* that iteratively updates the residual and selects experts through tensor operations. As detailed in Algorithm 1, this iterative approach generates an ordered contribution sequence of experts.

Search Efficiency. As derived in Appendix B.3, our greedy selection strategy achieves a total time complexity of $\mathcal{O}(L \cdot n \cdot N_e \cdot Bd)$, where L is the number of layers, n is the target number of experts, N_e is the total number of experts, and Bd represents the token-dimension product of the cached statistics. This provides a significant computational speedup over the evaluation of all $C(N_e, n)$ expert combinations, which is prohibitive for modern MoE models with large N_e. By decoupling the selection logic from the forward pass and operating on a linear search space, OMP-MoE can prune models with over 128 experts within minutes.

3.3 Cross-Layer Budget Allocation with Routing Stability

Uniform pruning assigns the same expert budget to each MoE layer, although different layers may have different reconstruction difficulty. Moreover, removing experts changes the normalization term of the gating softmax, leading to routing renormalization Dai et al. (2022). We therefore allocate experts across layers by jointly considering reconstruction error and preserved routing mass.

For layer l, OMP-MoE produces an ordered expert sequence $\pi_l = (e_{l,1}, e_{l,2}, \ldots, e_{l,N_e})$. If the first k experts are retained, the residual is

$$\mathbf{R}_l(k) = \mathbf{Y}_l - \sum_{t=1}^{k} \mathbf{V}_{l,e_{l,t}}, \tag{6}$$

and the normalized reconstruction error is

$$r_l(k) = \frac{\|\mathbf{R}_l(k)\|_F^2}{\|\mathbf{Y}_l\|_F^2}. \tag{7}$$

To measure the routing mass preserved by the retained experts, we accumulate the gating weight $m_{l,e}$ of expert e on the calibration set and normalize it as

$$\tilde{m}_{l,e} = \frac{m_{l,e}}{\sum_{j=1}^{N_e} m_{l,j}}. \tag{8}$$

The routing coverage of the first k retained experts is

$$C_l(k) = \sum_{t=1}^{k} \tilde{m}_{l,e_{l,t}}, \tag{9}$$

and we define the routing risk as

$$\psi_l(k) = -\log(C_l(k) + \zeta), \tag{10}$$

where ζ is a small constant for numerical stability. The layer-wise allocation cost is then

$$F_l(k) = r_l(k) + \lambda \psi_l(k), \tag{11}$$

where λ controls the strength of the routing-risk term.

Given a total expert budget C_{tot}, we choose the number of retained experts $\{k_l\}_{l=1}^{L}$ by solving

$$\min_{\{k_l\}_{l=1}^{L}} \sum_{l=1}^{L} F_l(k_l) \quad \text{s.t.} \quad \sum_{l=1}^{L} k_l = C_{\text{tot}}, \quad k_l \in \{1, \ldots, N_e\}. \tag{12}$$

We solve Eq. equation 12 with a discrete water-filling procedure. We initialize each layer with one retained expert and then allocate the remaining $C_{\text{tot}} - L$ experts one by one. At each step, we add one expert to the layer with the largest marginal cost decrease:

$$l^\star = \arg\max_{l:k_l<N_e} \Delta_l(k_l+1), \quad \Delta_l(k_l+1) = F_l(k_l) - F_l(k_l+1). \tag{13}$$

Table 1: Zero-shot performance comparison across 6 tasks on Qwen, DeepSeek, GPT-OSS and Mixtral MoE models at 25% and 50% expert pruning ratios. Bold indicates the best performance. 'Avg.' represents the average accuracy across 6 tasks.

Method	ARC-c	BoolQ	HellaS.	MMLU	OBQA	WinoG.	Avg.	ARC-c	BoolQ	HellaS.	MMLU	OBQA	WinoG.	Avg.
	Qwen3-30B-A3B							**DeepSeek-V2-Lite**						
Original	0.528	0.887	0.596	0.778	0.346	0.703	0.640	0.465	0.799	0.587	0.551	0.348	0.710	0.577
	Pruning Ratio 25%							Pruning Ratio 25%						
MC-SMoE	0.392	0.777	0.415	0.540	0.318	0.588	0.505	0.367	0.713	0.531	0.422	**0.366**	0.687	0.514
HC-SMoE	0.458	0.865	0.515	0.669	**0.410**	**0.704**	0.603	0.420	0.722	0.560	0.458	0.280	0.695	0.523
NAEE	0.481	0.870	0.555	0.701	0.290	0.693	0.598	0.375	0.669	0.531	0.365	0.290	0.669	0.483
DiEP	0.505	0.871	0.567	0.646	0.334	0.702	0.604	0.433	0.740	0.550	**0.505**	0.302	0.660	0.532
MoNE	0.523	0.871	0.580	0.730	0.308	0.684	0.616	**0.456**	0.720	0.576	0.474	0.314	**0.702**	0.540
Shapley-MoE	0.458	0.841	0.524	-	0.300	0.650	-	0.395	**0.760**	0.512	-	0.262	0.673	-
OMP-MoE (Ours)	**0.534**	**0.890**	**0.592**	**0.747**	0.338	0.699	**0.633**	**0.456**	0.713	**0.583**	0.494	0.320	0.697	**0.544**
	Pruning Ratio 50%							Pruning Ratio 50%						
MC-SMoE	0.195	0.583	0.273	0.261	0.250	0.470	0.339	0.288	0.595	0.438	0.237	0.306	0.598	0.410
HC-SMoE	0.358	0.830	0.418	0.461	**0.362**	0.652	0.513	0.325	0.564	0.456	**0.302**	0.216	0.625	0.415
NAEE	0.369	0.805	0.467	0.521	0.262	0.665	0.515	0.328	0.583	0.409	0.265	0.218	0.585	0.398
DiEP	0.389	0.762	0.486	0.299	0.244	0.634	0.469	0.294	**0.617**	0.441	0.256	0.212	0.569	0.398
MoNE	0.387	0.866	**0.553**	0.516	0.286	0.673	0.547	**0.375**	0.548	0.484	0.253	0.258	0.611	0.422
Shapley-MoE	0.320	0.712	0.420	-	0.222	0.591	-	0.268	0.240	0.395	-	0.240	0.603	-
OMP-MoE (Ours)	**0.533**	**0.879**	0.546	**0.605**	0.324	**0.695**	**0.597**	0.354	0.602	**0.513**	0.273	**0.310**	**0.631**	**0.447**
	GPT-OSS-20B							**Mixtral-8×7B**						
Original	0.451	0.757	0.415	0.566	0.27	0.657	0.519	0.565	0.851	0.649	0.671	0.350	0.759	0.641
	Pruning Ratio 25%							Pruning Ratio 25%						
MC-SMoE	0.399	0.725	0.399	0.488	0.258	**0.665**	0.489	0.262	0.521	0.432	0.250	0.194	0.585	0.374
HC-SMoE	0.285	0.622	0.336	0.425	0.182	0.617	0.411	0.450	0.830	0.570	0.560	0.290	0.745	0.574
NAEE	0.422	0.731	0.400	0.547	0.232	0.624	0.493	0.516	**0.836**	0.616	0.587	**0.330**	0.754	0.607
DiEP	0.379	0.751	0.393	0.527	0.224	0.593	0.478	0.514	0.835	0.612	0.598	0.302	0.740	0.600
MoNE	0.421	0.653	0.406	0.503	**0.268**	0.654	0.484	0.463	0.702	0.594	0.493	0.300	0.728	0.547
OMP-MoE (Ours)	**0.424**	**0.762**	**0.409**	**0.555**	0.260	0.650	**0.510**	**0.535**	0.831	**0.624**	**0.604**	**0.330**	**0.766**	**0.615**
	Pruning Ratio 50%							Pruning Ratio 50%						
MC-SMoE	**0.337**	0.716	0.365	0.376	0.236	0.635	0.444	0.212	0.495	0.277	0.245	0.108	0.496	0.306
HC-SMoE	0.208	0.421	0.31	0.295	0.174	0.555	0.327	0.322	0.754	0.493	0.392	0.256	0.671	0.481
NAEE	0.284	0.659	0.359	0.367	0.178	0.580	0.405	**0.489**	0.814	0.577	0.473	0.290	0.729	0.562
DiEP	0.277	0.49	0.336	0.315	0.194	0.549	0.360	0.473	0.812	**0.580**	0.489	0.292	**0.740**	0.564
MoNE	0.265	0.476	0.369	0.240	0.222	0.627	0.367	0.310	0.620	0.412	0.231	0.202	0.647	0.404
OMP-MoE (Ours)	0.333	**0.750**	**0.376**	**0.438**	**0.256**	**0.642**	**0.466**	0.482	**0.830**	0.577	**0.497**	**0.298**	0.728	**0.569**

The update is then $k_{l^\star} \leftarrow k_{l^\star} + 1$. This greedy allocation places each added expert in the layer where it gives the largest immediate decrease in the sum of reconstruction error and routing risk. More details are given in Appendix B.1.

3.4 OMP-MoE† : Adaptive Inference via Energy Prediction

To alleviate the compute bottleneck during deployment, we introduce OMP-MoE† as an adaptive inference strategy. This algorithm leverages the precomputed energy coefficients to skip unnecessary experts for each input token, thereby lowering the total computational overhead without sacrificing the fidelity of the reconstructed signal.

We precompute an energy coefficient $\alpha_e = \mathbb{E}[\|F_e(\mathbf{x})\|^2]$ which is set to quantify the potential reconstruction contribution of each expert in an one-time offline calibration phase. During inference, for a token $\mathbf{x}_i$ with gating weights $g_{i,j}$, we predict the contribution energy as $\hat{\mathcal{E}}_{i,j} = g_{i,j}^2 \cdot \alpha_{e_{i,j}}$. We then select the minimum number of experts k_i required to satisfy a relative residual threshold ϵ:

$$k_i = \min\{m \in [1, K] \mid (1 - \frac{\sum_{j=1}^{m} \hat{\mathcal{E}}_{i,j}}{\sum_{j=1}^{K} \hat{\mathcal{E}}_{i,j} + \zeta}) \leq \epsilon\}. \tag{14}$$

where ζ is a small constant to ensure numerical stability. Appendix A.5 further gives a gradient-sensitivity view showing that the energy rule controls the expected downstream loss perturbation by skipping low-energy expert directions. Under this dynamic scheme, we execute only the first k_i experts and treat the contributions of the remaining candidates as zero. As analysed in Appendix B.4, this mechanism yields a theoretical efficiency gain of $\eta = 1 - \mathbb{E}[m^*|\epsilon]/K$, where $\mathbb{E}[m^*|\epsilon]$ is the expected number of activated experts under threshold ϵ. Consequently, OMP-MoE† transforms

Table 2: Comparison of search efficiency and peak memory usage of NAEE, DiEP and OMP-MoE at a 50% pruning ratio. Measurements are taken on 8 × NVIDIA H20 GPUs.

Method	Qwen3-30B-A3B		DeepSeek-V2-Lite		GPT-OSS-20B		Mixtral-8×7B	
	Time (s)	Mem (GB)	Time (s)	Mem (GB)	Time (s)	Mem (GB)	Time (s)	Mem (GB)
NAEE	21301	79.4	8017	42.5	12787	66.3	7168	102.4
DiEP	8972	267.0	2745	98.8	4189	122.3	5957	242.3
OMP-MoE (Ours)	**641**	**75.0**	**358**	**40.6**	**275**	**56.8**	**188**	**97.5**

Table 3: Inference performance and efficiency trade offs on DeepSeek-V2-Lite model. We report the average zero-shot accuracy across 8 benchmarks with the total inference time, and the speedup at 25% and 50% pruning ratios. The analysis compares the static pruning of OMP-MoE against the OMP-MoE† algorithm with cross layer allocation enabled.

Prun. Ratio	OMP-MoE	OMP-MoE†	Avg. Acc	Cost ↓	Speedup ↑
0	-	-	0.609	2459s	1.00×
25%	✓	-	0.576	2222s	1.11×
25%	✓	✓	0.581	2019s	1.22×
50%	✓	-	0.489	1825s	1.35×
50%	✓	✓	0.476	1690s	1.46×

the rigid execution path of standard MoE into an input-aware mechanism, significantly reducing floating-point operations while preserving the reasoning fidelity of the model.

4 Experiments

We evaluate OMP-MoE across multiple MoE LLMs and conduct detailed ablation studies on its key design choices. Further implementation details, expanded results, and analyses are available in Appendix C, D.1, and D.4, respectively.

4.1 Experimental Setup

We conduct comprehensive evaluations across a diverse range of MoE architectures, including Qwen3-30B-A3B Yang et al. (2025), DeepSeek-V2-Lite Liu et al. (2024), GPT-OSS-20B Agarwal et al. (2025), and Mixtral-8×7B Jiang et al. (2024). Our OMP-MoE is benchmarked against state-of-the-art merging (MC-SMoE Li et al. (2024), HC-SMoE Chen et al. (2025)) and pruning baselines (NAEE Lu et al. (2024), MoNE Zhang et al. (2025), DiEP Bai et al. (2025), Shapley-MoE Huang et al. (2025)). To assess linguistic and reasoning proficiency, we evaluate zero-shot accuracy on eight general benchmarks: ARC-Challenge and ARC-Easy Clark et al. (2018), BoolQ Clark et al. (2019), HellaSwag Zellers et al. (2019), MMLU Hendrycks et al. (2021), OpenBookQA Mihaylov et al. (2018), RTE Bentivogli et al. (2009), and WinoGrande Sakaguchi et al. (2020). We further evaluate perplexity, compositional reasoning on BBH Suzgun et al. (2022), mathematical reasoning on GSM8K Cobbe et al. (2021), and long-context modeling on LongBench Bai et al. (2024). Implementation utilizes 64 WikiText-2 Merity et al. (2017) calibration samples with risk penalty coefficient $\lambda = 3.0$ and dynamic threshold $\epsilon = 0.1$ on 8× NVIDIA H20 GPUs.

4.2 Main Results

General Capabilities and Comparative Analysis. We evaluate the zero-shot performance of OMP-MoE across 6 tasks on Qwen, DeepSeek, GPT-OSS and Mixtral MoE model. Table 1 summarizes the average accuracy on these models at 25% and 50% expert pruning ratios. OMP-MoE consistently achieves superior accuracy compared to state-of-the-art merging and pruning baselines. At a 50% expert pruning ratio, OMP-MoE maintains an average accuracy of 0.597 for the Qwen3-30B-A3B model and preserves 93.3% of the original 0.640 baseline performance. Simultaneously, we observe similar patterns of resilience for other MoE models. These results confirm that our OMP-MoE effectively identifies the experts that are essential for model performance. We provide complete evaluation results across 8-task benchmarks in Appendix D.1. The robustness and scaling results in

Table 4: Evaluation of mathematical reasoning, long-context modeling, and compositional reasoning tasks on Qwen3-30B-A3B at a 50% pruning ratio.

Method	GSM8K	LongBench	BBH
Original	0.891	0.578	0.363
MC-SMoE	0.005	0.504	0.219
HC-SMoE	0.007	0.284	0.235
NAEE	0.236	0.470	0.310
MoNE	0.210	0.496	0.262
DiEP	0.007	0.329	0.299
OMP-MoE	**0.531**	**0.523**	**0.323**

Table 5: Evaluation of OMP-MoE combined with post-training quantization and pruning methods on DeepSeek-V2-Lite. We report the perplexity on WikiText-2 and C4 datasets and the average accuracy across 8 zero-shot tasks.

Method	WikiText-2	C4	Avg. Acc
OMP-MoE 25%	13.824	10.618	0.576
+ GPTQ	14.236	12.438	0.541
+ SparseGPT (4:8)	15.673	13.709	0.525
+ Wanda	14.430	13.321	0.534

Appendix D.7 further show that OMP-MoE remains effective under more aggressive compression and scales to larger models.

Pruning Overhead and Inference Acceleration. Table 2 reports the pruning overhead at a 50% pruning ratio. On Qwen3-30B-A3B, OMP-MoE finishes search in 641 seconds, achieving a 33× speedup over NAEE and a 14× speedup over DiEP, while using lower peak memory than both baselines. Similar gains are observed across DeepSeek-V2-Lite, GPT-OSS-20B, and Mixtral-8×7B, confirming the efficiency of the linear OMP search. Table 3 further shows that OMP-MoE reduces inference cost on DeepSeek-V2-Lite, with OMP-MoE† reaching a 1.46× speedup at a 50% pruning ratio by skipping low-energy expert executions.

Performance on Reasoning-Heavy and Long-Context Tasks. We evaluate OMP-MoE on GSM8K Cobbe et al. (2021), LongBench Bai et al. (2024), and BBH Suzgun et al. (2022) to test mathematical reasoning, long-context modeling, and compositional reasoning, which remain central in recent work on efficient reasoning Chen et al. (2026); Zhu et al. (2026b) and long-context modeling Hu et al. (2026). Table 4 presents the experimental results for the Qwen3-30B-A3B model. The model retains substantial reasoning ability even after removing 50% of the experts, achieving 0.531 on GSM8K, 0.523 on LongBench, and 0.323 on BBH. Compared with NAEE, OMP-MoE improves GSM8K from 0.236 to 0.531, showing that the selected experts are important for difficult reasoning and long-context inputs. Appendix D.2 adds direct comparisons with REAP Lasby et al. (2026) and EASY-EP Dong et al. (2025), while Appendix D.8 reports GSM8K and HumanEval Chen et al. (2021) under task-relevant calibration.

Orthogonality with Quantization and Pruning. We further combine 25% expert pruning with GPTQ Frantar et al. (2023), SparseGPT Frantar & Alistarh (2023), and Wanda Sun et al. (2024). Table 5 shows that OMP-MoE remains compatible with quantization and unstructured pruning techniques on DeepSeek-V2-Lite. After adding GPTQ, SparseGPT, or Wanda, the model still retains average accuracies of 0.541, 0.525, and 0.534, respectively, indicating that expert-level pruning can be combined with quantization and unstructured pruning for stronger compression.

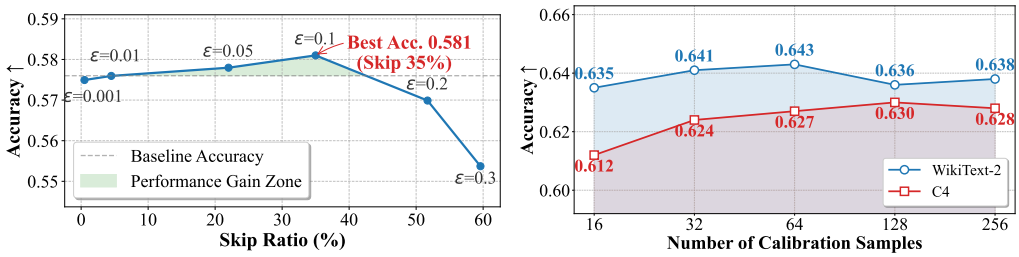
4.3 Ablation Studies

Effectiveness of Cross-Layer Budget Allocation. Table 6 compares allocation methods. At 50% pruning, OMP-MoE reaches 0.643 on Qwen3-30B-A3B, exceeding the 0.511 achieved by DiEP and the 0.480 achieved by HEAPr. Relative to uniform allocation, it improves Qwen3 from 0.618 to 0.643 and DeepSeek-V2-Lite from 0.468 to 0.489, confirming effective layer-wise budgeting. The consistent gains on both architectures indicate that layer sensitivity is heterogeneous and cannot be captured by assigning the same pruning ratio to every layer.

Table 6: Performance of different cross-layer allocation methods on 8 tasks zero-shot average accuracy.

Method	Qwen3-30B-A3B		DeepSeek-V2-Lite	
	25%	50%	25%	50%
Sub-MoE	0.601	0.574	-	-
HEAPr	0.590	0.480	-	-
HC-SMoE	0.644	0.536	0.552	0.456
DiEP	0.640	0.511	0.566	0.435
OMP-MoE				
w/o cross-layer	0.671	0.618	0.572	0.468
w cross-layer	**0.676**	**0.643**	**0.576**	**0.489**

Impact of Renormalization Risk Penalty. Table 7 studies the routing-risk weight λ at 50% pruning. Reconstruction error alone ($\lambda = 0$) yields 0.618 on Qwen3-30B-A3B, whereas including preserved



(a) Trade off between zero-shot accuracy and the expert skip ratio.

(b) Calibration sensitivity.

Figure 3: Ablations across eight benchmarks. (a) Accuracy and expert skipping under different adaptive thresholds on 25%-pruned DeepSeek-V2-Lite. (b) Accuracy under different calibration sizes and domains on 50%-pruned Qwen3-30B-A3B.

routing mass raises accuracy to 0.643. Thus, $\psi_l(k) = -\log(C_l(k) + \zeta)$ helps avoid allocations that induce strong routing renormalization. The monotonic improvement as λ increases further suggests that reconstruction fidelity and routing stability provide complementary signals for allocation.

Impact of OMP-MoE† Dynamic Threshold. The threshold ϵ controls the residual expert energy skipped at inference. In Figure 3(a), $\epsilon = 0.1$ attains the best accuracy of 0.581 while skipping about 35% of expert executions on DeepSeek-V2-Lite. Larger thresholds remove too much computation and reduce accuracy. This operating point therefore provides a favorable balance between predictive quality and inference efficiency without additional training.

Table 7: Performance of different routing-risk weights on Qwen3-30B-A3B at a 50% pruning ratio.

Risk Function $\psi_l(k)$	**Routing-Risk Weight** λ			
	0	1	2	3
$-\log(C_l(k) + \zeta)$	0.618	0.622	0.625	**0.643**

Impact of Calibration Configurations. Figure 3(b) studies calibration size and domain on 50%-pruned Qwen3-30B-A3B. WikiText-2 peaks at 0.643 with 64 samples, while C4 reaches 0.630 with 128. Additional samples provide little benefit. Appendix D.9 further shows a positive relation between normalized reconstruction fidelity and downstream accuracy. These results indicate that a modest calibration set is sufficient, although the calibration domain can affect the attainable accuracy.

5 Conclusion

Existing MoE compression methods rely on costly expert selection in a combinatorial search space and use uniform and fixed resource budgets across layers and inputs. This design ignores the structural heterogeneity of MoE layers and leads to suboptimal compression and unnecessary inference costs. We present OMP-MoE as a new framework for MoE compression that reformulates expert selection as a sparse signal reconstruction task. This formulation reduces the search space from combinatorial to linear and significantly accelerates the search process. We also introduce cross-layer budget allocation, which assigns compression budgets according to the redundancy of each layer, and OMP-MoE† adaptive inference, which dynamically adjusts computation for different inputs. These innovations improve compression effectiveness and reduce inference costs. Comprehensive experiments across diverse MoE models validate our framework. Its training-free operation and consistent effectiveness across architectures make OMP-MoE a practical solution for deploying large-scale MoE LLMs.

Limitations. Following most MoE compressors, our technique relies on calibration dataset selection, and we will explore advanced calibration procedures Shi et al. (2025) in future work. Future work could combine OMP-MoE with complementary compression and adaptation techniques Li et al. (2025c;b); Zhu et al. (2025; 2026b) to further improve efficiency while preserving domain-specific capabilities. Finally, the theoretical connection to OMP offers a promising basis for studying how expert selection relates to model interpretability and robustness under distribution shifts or perturbations.

AI USE STATEMENT

During the preparation of this manuscript, we used generative AI tools to improve grammar, phrasing, clarity, and readability, and to assist with manuscript review, numerical consistency checks, and feedback on mathematical claims and experimental reporting. The tools were not used to generate experimental data or results. All AI-assisted suggestions and edits were reviewed and verified by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human participants, human-subject experiments, or the collection of new personal data. We evaluate OMP-MoE using publicly available benchmark and calibration datasets and open-weight MoE language models, in accordance with their respective licenses and terms of use. Our work focuses on reducing expert redundancy and improving inference efficiency, and does not introduce a new high-stakes application.

REPRODUCIBILITY STATEMENT

The methodology section describes residual-based expert selection, cross-layer budget allocation, and adaptive expert execution in OMP-MoE. Algorithms 2 and 3 provide pseudocode for pruning and adaptive inference, respectively. Appendix C specifies model checkpoints, calibration data preparation, evaluation tasks, baseline configurations, hyperparameters, and the hardware and software environment. Appendix D provides task-level results, calibration and inference-threshold ablations, inference-time measurements, and additional robustness and scaling experiments. These descriptions and results support reproduction and comparison of the reported accuracy and efficiency measurements. Codes are available in Supplementary Material.

REFERENCES

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

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. Diep: Adaptive mixture-of-experts compression through differentiable expert pruning. *arXiv preprint arXiv:2509.16105*, 2025.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the Association for Computational Linguistics*, pp. 3119–3137, Bangkok, Thailand, Aug 2024. Association for Computational Linguistics.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST, 2009.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse moe via hierarchical clustering. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hslOzRxzXL.

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, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino,

Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Tianle Chen, Pengyu Cheng, Qiyuan Zhu, Jiacheng Wang, Bei Liu, Hao Gu, Ruijie Shen, Xiaofeng Hou, Sirui Han, and Jiacheng Liu. Adaptive spatial and temporal redundancy optimization for efficient reasoning in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24647–24662, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1130.

Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George Bm Van Den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc'Aurelio Ranzato, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In *Proceedings of the International Conference on Machine Learning*, volume 162, pp. 4057–4086. PMLR, 17–23 Jul 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the Association for Computational Linguistics*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

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

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

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the Association for Computational Linguistics*, pp. 7085–7095, Dublin, Ireland, May 2022. Association for Computational Linguistics.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Zican Dong, Han Peng, Peiyu Liu, Xin Zhao, Dong Wu, Feng Xiao, and Zhifeng Wang. Domain-specific pruning of large mixture-of-experts models with few-shot demonstrations. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 103552–103577, San Diego, CA, USA, 2025. Curran Associates, Inc.

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

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, volume 202, pp. 10323–10337. PMLR, Jul 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hR_SMu8cxCV.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Xiang Hu, Xinyu Wei, Hao Gu, Minshen Zhang, Tian Liang, Huayang Li, Lei Zhu, Yan Wang, Sirui Han, Yushi Bai, Kewei Tu, Haitao Mi, and Leo Liang. Hierarchical sparse attention done right: Toward infinite context modeling. *arXiv preprint arXiv:2607.02980*, 2026.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, Rongrong Ji, and Liujuan Cao. Discovering important experts for mixture-of-experts models pruning through a theoretical perspective. In *Advances in Neural Information Processing Systems*, 2025.

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

Junhyuck Kim, Jongho Park, Jaewoong Cho, and Dimitris Papailiopoulos. Lexico: Extreme KV cache compression via sparse coding over universal dictionaries. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Yh9vxlxnjA.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. In *International Conference on Learning Representations*, volume 2026, pp. 146883–146912, 2026.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Ke Li, Zheng Yang, Zhongbin Zhou, Feng Xue, Zhonglin Jiang, and Wenxiao Wang. Heapr: Hessian-based efficient atomic expert pruning in output space. *arXiv preprint arXiv:2509.22299*, 2025a.

Lujun Li, Dezhi Li, Cheng Lin, Wei Li, Wei Xue, Sirui Han, and Yike Guo. AIRA: Activation-informed low-rank adaptation for large models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1729–1739, Honolulu, HI, USA, 2025b. IEEE. doi: 10.1109/ICCV51701.2025.00169.

Lujun Li, Cheng Lin, Dezhi Li, You-Liang Huang, Wei Li, Tianyu Wu, Jie Zou, Wei Xue, Sirui Han, and Yike Guo. Efficient fine-tuning of large models via nested low-rank adaptation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22252–22262, Honolulu, HI, USA, 2025c. IEEE. doi: 10.1109/ICCV51701.2025.02066.

Lujun Li, Zhu Qiyuan, Jiacheng Wang, Wei Li, Hao Gu, Sirui Han, and Yike Guo. Sub-moe: Efficient mixture-of-expert llms compression via subspace expert merging. *arXiv preprint arXiv:2506.23266*, 2025d.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoe with hints from its routing policy. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eFWG9Cy3WK.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 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 ACL*, pp. 6159–6172, Bangkok, Thailand, Aug 2024. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, Oct 2018. Association for Computational Linguistics.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2020.

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

Weijie Shi, Jipeng Zhang, Yaguang Wu, Jingzhi Fang, Shibo Zhang, Yao Zhao, Hao Chen, Ruiyuan Zhang, Yue Cui, Jia Zhu, Sirui Han, Jiajie Xu, and Xiaofang Zhou. DIDS: Domain impact-aware data sampling for large language model training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4330–4350, Suzhou, China, Nov 2025. Association for Computational Linguistics.

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

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=PxoFut3dWW`.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

Binxing Xu, Hao Gu, Lujun Li, Hao Wang, Bei Liu, Jiacheng Liu, Qiyuan Zhu, Xintong Yang, Chao Li, Sirui Han, and Yike Guo. Bit-by-bit: Progressive QAT strategy with outlier channel splitting for stable low-bit LLMs. In *Proceedings of the 64th Annual Meeting of the ACL*, pp. 22284–22299, San Diego, California, United States, Jul 2026. Association for Computational Linguistics.

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

Jie Yang, Abdesselam Bouzerdoum, and Son Lam Phung. A neural network pruning approach based on compressive sampling. In *International Joint Conference on Neural Networks*, pp. 3428–3435, 2009.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, Jul 2019. Association for Computational Linguistics.

Geng Zhang, Yuxuan Han, Yuxuan Lou, Wangbo Zhao, Yiqi Zhang, and Yang You. Mone: Replacing redundant experts with lightweight novices for structured pruning of moe. *arXiv preprint arXiv:2507.00390*, 2025.

Qiyuan Zhu, Lujun Li, Dezhi Li, Jiacheng Liu, Pengyu Cheng, Yucheng Xu, Sirui Han, and Yike Guo. Outlier-aware model merging for efficient multitask inference. In *Proceedings of the ACM International Conference on Multimedia*, pp. 1032–1041, Dublin, Ireland, 2025. Association for Computing Machinery.

Qiyuan Zhu, Dezhi Li, Pengyu Cheng, Tianle Chen, Jiacheng Wang, Ruijie Shen, Hao Gu, Sida Lin, Zirui Liu, Jiacheng Liu, and Sirui Han. Fewer tokens, smaller cache: Reward-coordinated efficient reasoning. *arXiv preprint arXiv:2608.04771*, 2026a.

Qiyuan Zhu, Dezhi Li, Lujun Li, Xiaoyu Qin, Wei Li, Hao Gu, Hua Xu, Sirui Han, and Yike Guo. Outlier matters: Efficient long-to-short reasoning via outlier-guided model merging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):35213–35221, Jan 2026b.

A Formal Theoretical Analysis of OMP-MoE

In this section, we provide a rigorous mathematical treatment of the expert selection process. We frame the Mixture-of-Experts (MoE) pruning problem within the context of Hilbert space theory and prove the optimality of our greedy selection criterion under the linear decomposition property.

A.1 Mathematical Framework

Let $\mathcal{H} = \mathbb{R}^{B\times d}$ be the Hilbert space of tensors equipped with the Frobenius inner product $\langle \mathbf{A}, \mathbf{B}\rangle_{\mathcal{H}} = \text{Tr}(\mathbf{A}^\top \mathbf{B})$ and the induced norm $\|\mathbf{A}\|_{\mathcal{H}} = \sqrt{\langle \mathbf{A}, \mathbf{A}\rangle_{\mathcal{H}}}$.

In the standard Orthogonal Matching Pursuit (OMP) formulation, the objective is to approximate a signal $\mathbf{Y}$ using a linear combination of atoms from a dictionary $\mathcal{D} = \{\mathbf{V}_e\}_{e=1}^{N_e}$ with continuous coefficients $\mathbf{c} \in \mathbb{R}^{N_e}$. However, in the context of MoE pruning, "partial selection" of an expert is computationally invalid, because an expert is either executed (coefficient 1) or pruned (coefficient 0).

We formally define the *Expert Selection Problem* as a constrained optimization over the Boolean hypercube:

$$\min_{\mathbf{c}\in\{0,1\}^{N_e}} \left\| \mathbf{Y} - \sum_{e=1}^{N_e} c_e \mathbf{V}_e \right\|_{\mathcal{H}}^2 \quad \text{subject to} \quad \|\mathbf{c}\|_0 = n. \tag{15}$$

Let $E = \{e \mid c_e = 1\}$ be the support set. The problem reduces to finding the optimal subset indices:

$$\min_{E:|E|=n} \mathcal{J}(E) \coloneqq \left\| \mathbf{Y} - \sum_{e\in E} \mathbf{V}_e \right\|_{\mathcal{H}}^2. \tag{16}$$

A.2 Derivation of the Greedy Selection Rule

Unlike standard OMP, which requires an orthogonal projection of the signal onto the subspace spanned by the selected atoms (typically involving the inversion of the Gram matrix), our binary constraint simplifies the update dynamics significantly.

Theorem 1 (Optimality of Greedy Selection Criterion)**.** *Given a current residual $\mathbf{R}_t = \mathbf{Y} - \sum_{e\in E_t} \mathbf{V}_e$ at iteration t, and imposing the binary constraint $c_{e^*} \in \{0, 1\}$, the optimal next expert e^* to add to the support set E_t is determined by:*

$$e^* = \arg\max_{e\notin E_t} \left\{ 2\langle \mathbf{R}_t, \mathbf{V}_e\rangle_{\mathcal{H}} - \|\mathbf{V}_e\|_{\mathcal{H}}^2 \right\}. \tag{17}$$

Furthermore, the residual update is strictly additive: $\mathbf{R}_{t+1} = \mathbf{R}_t - \mathbf{V}_{e^*}$.

Proof. Let $e \notin E_t$ be a candidate expert. Under the binary constraint, if expert e is selected, its coefficient is fixed at 1. Thus, the new residual is deterministically defined as $\mathbf{R}_{t+1} = \mathbf{R}_t - \mathbf{V}_e$. We analyze the energy reduction (squared norm) of the residual:

$$\begin{aligned} \|\mathbf{R}_{t+1}\|_{\mathcal{H}}^2 &= \|\mathbf{R}_t - \mathbf{V}_e\|_{\mathcal{H}}^2 \\ &= \langle \mathbf{R}_t - \mathbf{V}_e, \mathbf{R}_t - \mathbf{V}_e\rangle_{\mathcal{H}} \\ &= \langle \mathbf{R}_t, \mathbf{R}_t\rangle_{\mathcal{H}} - 2\langle \mathbf{R}_t, \mathbf{V}_e\rangle_{\mathcal{H}} + \langle \mathbf{V}_e, \mathbf{V}_e\rangle_{\mathcal{H}} \\ &= \|\mathbf{R}_t\|_{\mathcal{H}}^2 - \underbrace{\left(2\langle \mathbf{R}_t, \mathbf{V}_e\rangle_{\mathcal{H}} - \|\mathbf{V}_e\|_{\mathcal{H}}^2\right)}_{\Delta(e;\mathbf{R}_t)}. \end{aligned} \tag{18}$$

Since $\|\mathbf{R}_t\|_{\mathcal{H}}^2$ is constant with respect to e, minimizing the new residual norm $\|\mathbf{R}_{t+1}\|_{\mathcal{H}}^2$ is equivalent to maximizing the term $\Delta(e; \mathbf{R}_t)$. This proves that the greedy selection based on Δ is locally optimal under the binary constraint. □

A.3 Comparison with Standard OMP Projection

It is crucial to distinguish this update rule from standard OMP. In standard OMP, the residual is updated as $\mathbf{R}_{t+1}^{std} = (\mathbf{I} - \mathbf{P}_{t+1})\mathbf{Y}$, where $\mathbf{P}_{t+1}$ is the orthogonal projector onto $\text{span}(\{\mathbf{V}_e\}_{e\in E_{t+1}})$.

This implies that previously selected experts re-adjust their coefficients to compensate for the new atom.

In OMP-MoE, the binary constraint forbids such re-adjustment (coefficients must remain 1). Consequently, the "orthogonalization" step is replaced by a direct subtraction. This yields two theoretical implications:

1. **Computational Efficiency:** We avoid the $\mathcal{O}(n^3)$ cost of matrix inversion required for projection, reducing the complexity to simple tensor additions.
2. **Independence of Selection:** The contribution of an expert $\mathbf{V}_e$ is invariant to the selection of future experts, preserving the physical interpretation of the MoE routing mechanism where expert weights are determined by the router, not by a post-hoc least-squares fit.

A.4 Budget Allocation via Marginal Gain

The cross-layer budget allocation involves distributing a total budget C_{tot} across L layers. Let $f_l(k) = \|\mathbf{R}_{l,k}\|_{\mathcal{H}}^2$ be the minimum residual norm for layer l using k experts.

Theorem 2 (Optimality of Greedy Allocation)**.** *If the sequence of marginal gains $\delta_l(k) = f_l(k-1) - f_l(k)$ is monotonically non-increasing for each layer l (i.e., f_l is discrete convex), then the greedy allocation strategy yields the globally optimal expert distribution $\{k_l^*\}_{l=1}^L$ that minimizes the total reconstruction error $\sum_{l=1}^L f_l(k_l)$.*

Proof. This follows from the theory of submodular optimization. The objective function $F(k_1, \ldots, k_L) = \sum f_l(k_l)$ is a separable function. If each f_l exhibits diminishing marginal returns (a property we empirically verify in Figure 1a), the greedy algorithm that at each step allocates one unit of budget to the layer $l = \arg\max \delta_l(k_l + 1)$ is guaranteed to reach the global optimum for the resource allocation problem under a matroid constraint. □

A.5 Reconstruction Error and Loss Perturbation

Let $\mathbf{Y}_l$ be the original output of layer l, and let the pruned or skipped output be

$$\tilde{\mathbf{Y}}_l = \mathbf{Y}_l - \mathbf{R}_l, \tag{19}$$

where $\mathbf{R}_l$ is the reconstruction residual. For a downstream loss $\mathcal{L}$, a local second-order expansion gives

$$\begin{aligned}\Delta\mathcal{L} &= \mathcal{L}(\{\tilde{\mathbf{Y}}_l\}_{l=1}^L) - \mathcal{L}(\{\mathbf{Y}_l\}_{l=1}^L) \\ &\approx -\sum_{l=1}^L \langle \nabla_{\mathbf{Y}_l}\mathcal{L}, \mathbf{R}_l\rangle \\ &\quad + \frac{1}{2}\sum_{l=1}^L \mathrm{vec}(\mathbf{R}_l)^\top \mathbf{H}_l \mathrm{vec}(\mathbf{R}_l).\end{aligned} \tag{20}$$

Assume bounded local sensitivity:

$$\|\nabla_{\mathbf{Y}_l}\mathcal{L}\|_F \le G_l, \qquad \|\mathbf{H}_l\|_2 \le \beta_l. \tag{21}$$

Then

$$|\Delta\mathcal{L}| \le \sum_{l=1}^L G_l\|\mathbf{R}_l\|_F + \frac{1}{2}\sum_{l=1}^L \beta_l\|\mathbf{R}_l\|_F^2. \tag{22}$$

Thus, reducing the layer-wise reconstruction residual reduces a local upper bound on the downstream loss perturbation.

We next apply this view to OMP-MoE†. For token $\mathbf{x}_i$, let the skipped perturbation be

$$\delta\mathbf{y}_i = \sum_{j > m_i^*} g_{i,j} F_{e_{i,j}}(\mathbf{x}_i). \tag{23}$$

With the calibrated expert energy

$$\alpha_e \approx \mathbb{E}_{\mathbf{x}} \|F_e(\mathbf{x})\|_2^2, \tag{24}$$

OMP-MoE† predicts the contribution energy as

$$\hat{\mathcal{E}}_{i,j} = g_{i,j}^2 \alpha_{e_{i,j}}. \tag{25}$$

Under a weak-correlation approximation among skipped expert outputs, the skipped output energy is bounded by

$$\mathbb{E}\|\delta \mathbf{y}_i\|_2^2 \lesssim \sum_{j > m_i^*} \hat{\mathcal{E}}_{i,j}. \tag{26}$$

Since OMP-MoE† chooses the smallest m_i^* satisfying

$$\frac{\sum_{j>m_i^*} \hat{\mathcal{E}}_{i,j}}{\sum_{j=1}^{K} \hat{\mathcal{E}}_{i,j} + \zeta} \leq \epsilon, \tag{27}$$

we obtain

$$\mathbb{E}\|\delta \mathbf{y}_i\|_2^2 \lesssim \epsilon \sum_{j=1}^{K} \hat{\mathcal{E}}_{i,j}. \tag{28}$$

Applying the local sensitivity bound at the token level gives

$$\mathbb{E}|\Delta \mathcal{L}_i| \lesssim C_{1,i}\sqrt{\epsilon} + C_{2,i}\epsilon, \tag{29}$$

where $C_{1,i}$ and $C_{2,i}$ absorb the local sensitivity constants and the total predicted expert energy. This result explains why the energy coefficient can skip low-energy expert directions while keeping the induced loss perturbation controlled.

Algorithm 2 Comprehensive OMP-MoE Pruning Pipeline

Require: MoE Model $\mathcal{M}$, Calibration manifold $\mathcal{D}$, Global pruning ratio P
Ensure: Optimal expert subsets $\{E_l\}_{l=1}^{L}$

Phase 1: Statistics Acquisition
for each layer $l \in \{1, \ldots, L\}$ **do**
    Execute a singular forward pass over $\mathcal{D}$ to populate the contribution cache
    Cache output signal $\mathbf{Y}_l$ and expert contribution atoms $\{\mathbf{V}_{l,e}\}_{e=1}^{N_e}$
    Precompute squared Frobenius norms $v_{l,e} = \|\mathbf{V}_{l,e}\|_F^2$ for all e
end for
Phase 2: Layer-wise Greedy Expert Selection
for each layer $l \in \{1, \ldots, L\}$ **do**
    Initialize residual signal $\mathbf{R}_0 = \mathbf{Y}_l$ and selected set $E_{l,0} = \emptyset$
    for iteration $t = 1$ to N_e **do**
        Identify $e^* = \arg\max_{e \notin E_{l,t-1}} (2\langle \mathbf{R}_{t-1}, \mathbf{V}_{l,e} \rangle_{\mathcal{H}} - v_{l,e})$
        Update residual: $\mathbf{R}_t = \mathbf{R}_{t-1} - \mathbf{V}_{l,e^*}$
        Append e^* to importance sequence π_l and record residual energy $r_l(t) = \|\mathbf{R}_t\|_F^2$
    end for
end for
Phase 3: Risk-Aware Global Budget Allocation
Distribute expert quotas $\{k_l\}_{l=1}^{L}$ by solving the constrained optimization in Eq. 12
Utilize a greedy water-filling strategy guided by marginal gain curves $\{r_l(k)\}$
Return Retained expert indices $\{E_l\}_{l=1}^{L}$ where $E_l = \{e \in \pi_l \mid \text{rank}(e) \leq k_l\}$

B Extended Algorithmic Procedures and Computational Analysis

In this section, we provide the comprehensive procedural logic for the OMP-MoE framework and conduct a formal analysis of its computational complexity. We delineate the transition from offline statistics collection to online adaptive inference, establishing the theoretical foundations for the efficiency gains reported in our experiments.

B.1 Comprehensive Pruning and Selection Pipeline

The OMP-MoE pruning pipeline is structured into three distinct phases: signal acquisition, greedy expert pursuit, and global resource allocation. Unlike traditional pruning methods that necessitate iterative model evaluations, our approach isolates the selection logic from the forward pass by operating on cached contribution atoms. The detailed implementation is provided in Algorithm 2.

Algorithm 3 Adaptive Inference Mechanism (OMP-MoE†)

Require: Token input $\mathbf{x}$, Top-K expert indices $\{e_1, \ldots, e_K\}$, Energy coefficients $\{\alpha_e\}$, Residual threshold ϵ

Ensure: Parsimonious output signal $\mathbf{y}$

1: Retrieve gating weights g_j for $j \in \{1, \ldots, K\}$

2: Estimate contribution energy: $\hat{\mathcal{E}}_j = g_j^2 \cdot \alpha_{e_j}$

3: Compute prefix sums of predicted energy: $S_m = \sum_{j=1}^{m} \hat{\mathcal{E}}_j$

4: Determine minimal expert cardinality: $m^* = \min\{m \in [1, K] \mid (1 - \frac{S_m}{S_K + \zeta}) \leq \epsilon\}$

5: Execute FFN sub-networks only for the subset $\{e_1, \ldots, e_{m^*}\}$

6: **Return** Aggregated output $\mathbf{y} = \sum_{j=1}^{m^*} g_j F_{e_j}(\mathbf{x})$

B.2 Adaptive Online Inference via OMP-MoE†

To further catalyze inference-time efficiency, we introduce OMP-MoE† , an adaptive mechanism that dynamically modulates the expert activation count for each input token. By leveraging the precomputed energy coefficients α_e, the model predicts the reconstruction sufficiency of the top-K candidates before full execution. This mechanism is formalized in Algorithm 3.

B.3 Computational Complexity and Asymptotic Bounds

We conduct a rigorous evaluation of the computational trajectory associated with OMP MoE by examining the asymptotic scaling across two distinct temporal phases. Let the configuration space be defined by the number of layers L, the total expert count per layer N_e, the calibration batch size B, and the hidden dimensionality d.

The Statistics Acquisition phase necessitates a singular forward evaluation of the model over the calibration manifold. The computational overhead of this phase is isomorphic to standard sparse MoE inference: $\mathcal{O}(L \cdot B \cdot K \cdot d_{ffn})$, where K denotes the active expert count and d_{ffn} represents the intermediate dimension of the feed-forward network. The memory footprint required to cache the contribution atoms is $\mathcal{O}(L \cdot N_e \cdot B \cdot d)$. This cost is incurred exclusively once per architecture and remains invariant to the number of pruning ratios under evaluation.

The Recursive Selection complexity describes the iterative greedy pursuit within a non-orthogonal dictionary of expert contribution atoms. For a given layer, the selection of n experts involves n iterations of inner product computations. The total complexity is expressed as:

$$\mathcal{C}_{\text{search}} \approx \sum_{l=1}^{L} \sum_{t=1}^{n} (N_e - t + 1) \cdot Bd = \mathcal{O}(L \cdot n \cdot N_e \cdot Bd). \tag{30}$$

This linear scaling with respect to the token batch B ensures that OMP MoE remains computationally tractable even as N_e increases. This property provides a significant advantage over methods requiring combinatorial forward passes. Table 8 provides a comparative analysis of these complexity bounds against prevailing baselines.

As summarized in Table 8, OMP-MoE replaces the repeated forward evaluations or gradient steps of the baselines with a single statistics-acquisition pass followed by operations on cached atoms. Its search is polynomial in the number of experts and retained experts, whereas the subset enumeration term in NAEE grows combinatorially.

Table 8: Asymptotic complexity comparison between OMP MoE and state of the art pruning methodologies. We denote N_{iter} as the number of iterations required for combinatorial search and T_{ft} as the duration of the fine-tuning stage.

Method	Search Complexity	Forward Passes	Optimization Type
NAEE	$\mathcal{O}(L \cdot N_{iter} \cdot \binom{N_e}{n})$	Massive	Combinatorial
DiEP	$\mathcal{O}(T_{ft} \cdot L \cdot B \cdot d^2)$	Extensive	Gradient-based
OMP-MoE	$\mathcal{O}(L \cdot n \cdot N_e \cdot Bd)$	**Singular**	Greedy Pursuit

B.4 Theoretical FLOPs Reduction in OMP-MoE†

We quantify the routed-expert FLOPs per token in one MoE layer, counting a multiply-add as two FLOPs. For the gated FFNs in Table 9, the gate, up, and down projections have a leading cost of $\mathcal{F}_{\text{expert}} = 6d\, d_{\text{ffn}}$ per expert. The baseline layer cost is $\mathcal{F}_{\text{baseline}} = \mathcal{F}_{\text{fixed}} + K\mathcal{F}_{\text{expert}}$, where $\mathcal{F}_{\text{fixed}}$ includes routing and any shared-expert computation. Element-wise FFN operations are omitted from this leading-order count.

Under OMP-MoE†, let $\mathbb{E}[m^* \mid \epsilon]$ be the mean number of executed routed experts. Using the same per-expert cost, the expected layer FLOPs are

$$\mathcal{F}_{\text{adaptive}}(\epsilon) = \mathcal{F}_{\text{fixed}} + \mathcal{F}_{\text{selection}} + \mathbb{E}[m^* \mid \epsilon]\mathcal{F}_{\text{expert}}, \tag{31}$$

where $\mathcal{F}_{\text{selection}}$ denotes the additional energy-prediction and expert-selection overhead. With $m^* = K$, the routed-expert cost equals the baseline routed-expert cost. For Mixtral with $K = 2$, executing one expert instead of two reduces this component by 50%. Across tokens, the routed-expert FLOPs reduction is $\eta = 1 - \mathbb{E}[m^* \mid \epsilon]/K$. Table 9 reports this reduction and the corresponding normalized routed-expert cost. These quantities exclude routing, selection overhead, shared experts, and attention, and therefore do not represent end-to-end FLOPs reductions.

Table 9: Normalized routed-expert FLOPs per token in one MoE layer under OMP-MoE† with $\epsilon = 0.1$. Each baseline routed-expert cost is normalized to 1. The adaptive cost is $\mathbb{E}[m^* \mid \epsilon]/K$.

Architecture	Active Experts (K)	$\mathbb{E}[m^*\|\epsilon]$	Relative FLOPs ↓	Efficiency Gain (η)
Mixtral-8×7B	2	1.22	1.000 → 0.610	39.0%
DeepSeek-V2-Lite	6	3.90	1.000 → 0.650	35.0%
Qwen3-30B-A3B	8	5.14	1.000 → 0.643	35.8%

Table 9 predicts routed-expert FLOPs reductions between 35.0% and 39.0% across the three architectures. The similar range despite different native top-K values indicates that the adaptive criterion scales with the fraction of contribution energy captured, rather than relying on a particular expert count.

C Implementation and Configuration

In this section, we provide exhaustive implementation details, model specifications, and hyperparameter configurations to ensure the reproducibility of our results.

C.1 Model Architecture Specifications

Table 10 provides the detailed architectural parameters for the models evaluated in this work. We also include a detailed breakdown of routing mechanisms in Table 11.

We use the exact checkpoints `Qwen/Qwen3-30B-A3B`, `deepseek-ai/DeepSeek-V2-Lite`, `openai/gpt-oss-20b`, and `mistralai/Mixtral-8x7B-v0.1` in HuggingFace. As formalized in Eq. 2, OMP-MoE consumes the weighted expert contributions produced by each checkpoint's architecture-native router. The selection objective is therefore unchanged across softmax-, sigmoid-, or otherwise transformed router scores, provided that the native score transformation, top-k selection, and normalization are retained when constructing $G(\mathbf{X})$.

We do not enable explicit chain-of-thought reasoning in any generative evaluation. For Qwen3, we explicitly set `enable_thinking=False`. Calibration and evaluation formatting are distinct: calibration always uses directly tokenized raw text without a chat template, whereas HumanEval uses the `humaneval_instruct` task with `apply_chat_template` so that the prompt format matches the evaluated model. This evaluation-time template is not applied to calibration data.

Table 10: Detailed architectural specifications of the evaluated MoE models.

Model	Total Params	Active Params	Layers (L)	Experts (N_e)	Active (K)	Hidden Size (d)	Expert Intermediate	Dense Intermediate
`Qwen/Qwen3-30B-A3B`	30.5B	3.3B	48	128	8	2048	768	6144
`deepseek-ai/DeepSeek-V2-Lite`	15.7B	2.4B	27	64	6	2048	1408	10944
`openai/gpt-oss-20b`	21.0B	3.6B	24	32	4	2880	2880	–
`mistralai/Mixtral-8x7B-v0.1`	46.7B	12.9B	32	8	2	4096	14336	–

The values in Table 10 are read directly from each checkpoint's `config.json`: `hidden_size` gives the hidden size, `moe_intermediate_size` (or the expert `intermediate_size` field for Mixtral) gives the routed-expert FFN width, and a distinct dense/shared FFN width is reported separately when the configuration provides one. A dash indicates that the checkpoint has no separate dense intermediate-size field.

Table 11: Detailed routing mechanisms and gating configurations.

Model	Top-K	Shared Experts
Qwen3-30B-A3B	8	No
DeepSeek-V2-Lite	6	Yes
GPT-OSS-20B	4	No
Mixtral-8×7B-v0.1	2	No

Together, Tables 10 and 11 cover expert dictionaries ranging from 8 to 128 experts and native activation counts from 2 to 8. DeepSeek-V2-Lite additionally contains shared experts, which remain unchanged during pruning. This architectural diversity supports evaluating OMP-MoE on weighted contribution atoms instead of assuming a single router implementation.

C.2 Dataset Processing and Calibration

We use C4 and WikiText-2 for general-domain calibration and perplexity evaluation. For calibration, we extract raw text and tokenize it directly without applying a chat template. We concatenate the resulting token sequences into one token stream, divide that stream into non-overlapping fixed-length blocks of 4,096 tokens, retain the first 64 complete blocks, and discard any incomplete remainder. Thus, each calibration set contains exactly 262,144 tokens. These blocks are used to cache expert contribution atoms and total layer outputs for the OMP statistics-collection phase. The same preprocessing protocol is used for both corpora.

For the domain-specific generative experiments in Appendix D.8, we additionally use `allenai/tulu-3-sft-personas-math` for GSM8K and `theblackcat102/evol-codealpaca-v1` for HumanEval. For each domain-specific calibration set, we shuffle the training split with seed 42 and construct 64 sequences of 4,096 tokens, again totaling 262,144 tokens. We use raw-token inputs without chat templates during calibration, and OMP-MoE, REAP, and EASY-EP receive identical calibration sequences.

Table 12 details the characteristics and processing configurations of these datasets. We also provide the breakdown of zero-shot evaluation metrics in Table 13.

Together, Tables 12 and 13 distinguish the fixed calibration-token budget from downstream evaluation. Both calibration corpora use the same non-overlapping 64-block construction, while the evaluation tasks retain their task-specific metrics and shot settings.

C.3 Benchmark Evaluation Details

We evaluate the four MoE models across a total of twelve diverse benchmarks. These models include Qwen3-30B-A3B Yang et al. (2025), DeepSeek-V2-Lite Liu et al. (2024), GPT-OSS-20B Agarwal

Table 12: Characteristics and processing configurations of the calibration datasets. C4 and WikiText-2 are also used for perplexity evaluation.

Dataset	Domain	Processing Method	Block Length	Blocks
C4	Web Corpus	Non-overlapping fixed-length chunking	4096	64
WikiText-2	Wikipedia	Non-overlapping fixed-length chunking	4096	64
Tulu-3 SFT Personas Math	Mathematics	Raw-token fixed-length construction	4096	64
Evol-CodeAlpaca-v1	Code	Raw-token fixed-length construction	4096	64

Table 13: Benchmark evaluation tasks and their corresponding performance metrics.

Category	Task	Evaluation	Primary Metric
Perplexity	WikiText-2, C4	0-shot	Perplexity ↓
8 General Tasks	ARC-Challenge, ARC-Easy, BoolQ, HellaSwag, MMLU, OpenBookQA, RTE, WinoGrande	0-shot	Accuracy ↑
Compositional Reasoning	BBH	0-shot	Exact Match ↑
Mathematics	GSM8K	5-shot	Exact Match ↑
Long-context	LongBench	0-shot	Score ↑
Code Generation	HumanEval	0-shot	pass@1 ↑

et al. (2025), and Mixtral-8×7B Jiang et al. (2024). We report zero-shot accuracy on eight general benchmarks: ARC-Challenge and ARC-Easy Clark et al. (2018), BoolQ Clark et al. (2019), HellaSwag Zellers et al. (2019), MMLU Hendrycks et al. (2021), OpenBookQA Mihaylov et al. (2018), RTE Bentivogli et al. (2009), and WinoGrande Sakaguchi et al. (2020).

We also conduct evaluations on specialized reasoning, long-context, and code-generation tasks. Table 13 summarizes the evaluation settings and primary metrics for these benchmarks. The Big Bench Hard (BBH) Suzgun et al. (2022) dataset tests compositional reasoning capabilities through zero-shot evaluation. We report the exact match score for the BBH benchmark. The GSM8K Cobbe et al. (2021) dataset assesses mathematical reasoning through 5-shot evaluation. We report the exact match score for this mathematical task. The LongBench Bai et al. (2024) benchmark evaluates long-context modeling proficiency through zero-shot evaluation. We report its comprehensive score metric. HumanEval Chen et al. (2021) evaluates code-generation correctness in the zero-shot setting, for which we report pass@1.

For the domain-specific Qwen3-30B-A3B experiments, GSM8K is evaluated in the 5-shot setting. HumanEval is evaluated with the `humaneval_instruct` task in `lm-eval`, using the model's chat template, `enable_thinking=False`, greedy decoding, `max_new_tokens=1024`, and batch size 8. The chat template is used for HumanEval evaluation only, and calibration remains based on the raw-token protocol described above.

C.4 Standardization of Router Renormalization

After experts are removed, we preserve each architecture's native router transformation and renormalization rule rather than imposing a universal softmax. In the notation of Eq. 2, pruning changes only the set of available routed experts. The score transformation, top-k selection, normalization, and any architecture-specific bias correction remain those of the original checkpoint. All pruning and merging baselines receive the same architecture-native treatment. Consequently, OMP-MoE operates on the resulting weighted contribution atoms regardless of whether the router begins from softmax, sigmoid, or another native score transformation.

C.5 Hyperparameter Display

Table 14 lists the hyperparameters used for the OMP-MoE pruning and OMP-MoE† inference stages. We specify the search and statistics collection configurations in Table 15.

For the combinatorial search baseline NAEE Lu et al. (2024), the search space grows factorially with the number of experts. To ensure computational feasibility on models with large expert counts, we imposed a computational budget cap. Specifically, we limited the search to a maximum of 200 iterations per layer for all models except Mixtral-8×7B, where the full search space remains tractable. For all other baselines, we strictly adhered to the hyperparameter configurations reported in their original papers.

Table 14: Hyperparameter settings for OMP-MoE pruning and OMP-MoE† inference.

Subgroup	Parameter	Value
OMP Search	risk penalty coefficient λ	3.0
	Numerical stability ζ	1e-6
Global Allocation	Min expert ratio	0.375
	Max expert ratio	0.875
OMP-MoE†	Adaptive threshold epsilon	0.1
	Numerical stability ζ	1e-6

Table 15: Search and statistics collection configurations.

Parameter	Configuration
Calibration Dataset	C4 and WikiText-2
Calibration Samples per Layer	64 samples
Search Batch Size	1
Vectorization	PyTorch torch.matmul
Layer Offloading	Enabled

C.6 Baseline Configuration Details

For baselines, MC-SMoE, NAEE, MoNE, and Shapley-MoE use a uniform pruning ratio across MoE layers, while DiEP, HC-SMoE, HEAPr, and Sub-MoE include cross-layer budget allocation. For NAEE, the combinatorial search space becomes infeasible on models with large expert counts. Therefore, we impose a fixed computational budget and cap its search to 200 iterations per layer for all models except Mixtral-8×7B, where the full search space is still tractable. This setting follows the same evaluation budget used for the timing comparison.

For the additional REAP Lasby et al. (2026) and EASY-EP Dong et al. (2025) comparisons, we use the same Qwen3-30B-A3B checkpoint, pruning ratios, evaluation harness, architecture-native router treatment, and calibration sequences as OMP-MoE. In particular, all three methods receive the same 64-sequence calibration budget and no chat template is applied during calibration. The reported values are reproduced under this shared protocol rather than copied from prior papers.

C.7 Hardware and Software Environment

Table 16 summarizes the hardware and software environments used for all experiments.

Table 16: Hardware and software environment configurations.

Category	Component	Specification
Hardware	GPU	8 × NVIDIA H20 (96GB VRAM)
	CPU	Intel(R) Xeon(R) Platinum 8358 @ 2.60GHz
	RAM	1TB DDR4
Software	OS	Ubuntu 22.04.3 LTS
	Python	3.10.12
	PyTorch	2.9.1+cu124
	Transformers	4.57.3

Table 16 fixes the compute and software stack used for both pruning-time and inference-time measurements. Reporting the accelerator count, memory capacity, and library versions is necessary for interpreting the absolute search and latency values in the subsequent experiments.

D Extended Experimental Results

This section expands the aggregate results in the main text with task-level comparisons, compatibility and sensitivity studies, inference measurements, stronger compression settings, scaling results, and specialized generative evaluations. Each subsection states the evaluation question and discusses the corresponding tables.

D.1 Detailed Results for Zero-shot Tasks

We provide the complete zero-shot evaluation results for 8 diverse tasks on Qwen3-30B-A3B, DeepSeek-V2-Lite, GPT-OSS-20B, and Mixtral-8×7B models at pruning ratios of 25% and 50%. The benchmarks encompass ARC challenge Clark et al. (2018), ARC easy, BoolQ Clark et al. (2019), HellaSwag Zellers et al. (2019), MMLU Hendrycks et al. (2021), OpenBookQA Mihaylov et al. (2018), RTE Bentivogli et al. (2009), and WinoGrande Sakaguchi et al. (2020). Table 17 and Table 18 list the average accuracy scores. We additionally report the variance for each performance metric to demonstrate the statistical stability and reliability of the selection results.

Table 17: Comprehensive zero-shot performance comparison on Qwen3-30B-A3B and DeepSeek-V2-Lite model. We report accuracy across eight benchmarks at 25% and 50% expert pruning ratios, corresponding to retained expert counts of Num=96 and Num=64 for Qwen3, with analogous settings for DeepSeek. Bold indicates the best performance among pruning methods.

Expert	Method	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.↑
					Qwen3-30B-A3B					
Num=128	Original	0.528	0.792	0.887	0.596	0.778	0.346	0.823	0.703	0.682
Num=96	MC-SMoE	0.392	0.647	0.777	0.415	0.540	0.318	**0.823**	0.588	0.562
	HC-SMoE	0.458	0.760	0.865	0.515	0.669	**0.410**	0.769	**0.704**	0.644
	NAEE	0.481	0.769	0.870	0.555	0.701	0.290	0.773	0.693	0.642
	DiEP	0.505	0.774	0.871	0.567	0.646	0.334	0.718	0.702	0.640
	MoNE	0.523	0.782	0.871	0.580	0.730	0.308	0.776	0.684	0.657
	Shapley-MoE	0.458	0.712	0.841	0.524	-	0.300	-	0.650	-
	OMP (Ours)	**0.534±0.008**	**0.803±0.015**	**0.890±0.005**	**0.592±0.005**	**0.747±0.003**	0.338±0.021	0.805±0.024	0.699±0.013	**0.676±0.005**
Num=64	MC-SMoE	0.195	0.309	0.583	0.273	0.261	0.250	0.729	0.470	0.384
	HC-SMoE	0.358	0.652	0.830	0.418	0.461	**0.362**	0.556	0.652	0.536
	NAEE	0.369	0.656	0.805	0.467	0.521	0.262	0.614	0.665	0.545
	DiEP	0.389	0.678	0.762	0.486	0.299	0.244	0.599	0.634	0.511
	MoNE	0.387	0.710	0.866	**0.553**	0.516	0.286	0.729	0.673	0.590
	Shapley-MoE	0.320	0.512	0.712	0.420	-	0.222	-	0.591	-
	OMP (Ours)	**0.533±0.009**	**0.795±0.015**	**0.879±0.006**	0.546±0.005	**0.605±0.004**	0.324±0.021	**0.765±0.026**	**0.695±0.013**	**0.643±0.005**
					DeepSeek-V2-Lite					
Num=64	Original	0.465	0.784	0.799	0.587	0.551	0.348	0.625	0.710	0.609
Num=48	MC-SMoE	0.367	0.608	0.713	0.531	0.422	**0.366**	0.585	0.687	0.535
	HC-SMoE	0.420	0.732	0.722	0.560	0.458	0.280	0.549	0.695	0.552
	NAEE	0.375	0.722	0.669	0.531	0.365	0.290	0.549	0.669	0.521
	DiEP	0.433	0.747	0.740	0.550	**0.505**	0.302	**0.588**	0.660	0.566
	MoNE	**0.456**	**0.774**	0.720	0.576	0.474	0.314	0.549	**0.702**	0.571
	Shapley-MoE	0.395	0.677	**0.760**	0.512	-	0.262	-	0.673	-
	OMP (Ours)	**0.456±0.009**	**0.774±0.015**	0.713±0.008	**0.583±0.005**	0.494±0.004	0.320±0.021	0.574±0.029	0.697±0.013	**0.576±0.005**
Num=32	MC-SMoE	0.288	0.366	0.595	0.438	0.237	0.306	0.516	0.598	0.418
	HC-SMoE	0.325	0.593	0.564	0.456	**0.302**	0.216	0.563	0.625	0.456
	NAEE	0.328	0.620	0.583	0.409	0.265	0.218	0.567	0.585	0.447
	DiEP	0.294	0.589	**0.617**	0.441	0.256	0.212	0.498	0.569	0.435
	MoNE	**0.375**	0.644	0.548	0.484	0.253	0.258	**0.570**	0.611	0.468
	Shapley-MoE	0.268	0.491	0.240	0.395	-	0.240	-	0.603	-
	OMP (Ours)	0.354±0.010	**0.679±0.014**	0.602±0.008	**0.513±0.005**	0.273±0.004	**0.310±0.020**	0.549±0.030	**0.631±0.014**	**0.489±0.005**

Tables 17 and 18 show that OMP-MoE obtains the highest average accuracy among the compared pruning methods for every evaluated architecture at both pruning ratios. The advantage is particularly visible at 50% pruning: OMP-MoE reaches 0.643 on Qwen3, 0.489 on DeepSeek-V2-Lite, 0.515 on

Table 18: Comprehensive zero-shot performance comparison on GPT-OSS-20B and Mixtral-8×7B model. We report accuracy across eight benchmarks at 25% and 50% expert pruning ratios, corresponding to retained expert counts of Num=24 and Num=16 for GPT-OSS, with analogous settings for Mixtral. Bold indicates the best performance among pruning methods.

Expert	Method	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.↑
GPT-OSS-20B										
Num=32	Original	0.451	0.775	0.757	0.415	0.566	0.270	0.700	0.657	0.574
Num=24	MC-SMoE	0.399	0.739	0.725	0.399	0.488	0.258	0.617	**0.665**	0.536
	HC-SMoE	0.285	0.592	0.622	0.336	0.425	0.182	0.610	0.617	0.459
	NAEE	0.422	0.729	0.731	0.400	0.547	0.232	0.664	0.624	0.544
	DiEP	0.379	0.712	0.751	0.393	0.527	0.224	**0.736**	0.593	0.539
	MoNE	0.421	0.745	0.653	0.406	0.503	**0.268**	0.625	0.654	0.534
	OMP (Ours)	**0.424±0.009**	**0.765±0.014**	**0.762±0.008**	**0.409±0.005**	**0.555±0.004**	0.260±0.020	0.693±0.029	0.650±0.013	**0.565±0.005**
Num=16	MC-SMoE	**0.337**	0.680	0.716	0.365	0.376	0.236	0.621	0.635	0.496
	HC-SMoE	0.208	0.440	0.421	0.310	0.295	0.174	0.599	0.555	0.375
	NAEE	0.284	0.568	0.659	0.359	0.367	0.178	0.599	0.580	0.449
	DiEP	0.277	0.560	0.490	0.336	0.315	0.194	0.585	0.549	0.413
	MoNE	0.265	0.583	0.476	0.369	0.240	0.222	0.527	0.627	0.414
	OMP (Ours)	0.333±0.009	**0.699±0.014**	**0.750±0.008**	**0.376±0.005**	**0.438±0.004**	**0.256±0.019**	**0.625±0.029**	**0.642±0.013**	**0.515±0.005**
Mixtral-8×7B										
Num=8	Original	0.565	0.842	0.851	0.649	0.671	0.350	0.711	0.759	0.675
Num=6	MC-SMoE	0.262	0.556	0.521	0.432	0.250	0.194	0.527	0.585	0.416
	HC-SMoE	0.450	0.730	0.830	0.570	0.560	0.290	**0.690**	0.745	0.608
	NAEE	0.516	**0.819**	**0.836**	0.616	0.587	**0.330**	0.679	0.754	0.642
	DiEP	0.514	0.809	0.835	0.612	0.598	0.302	0.657	0.740	0.633
	MoNE	0.463	0.775	0.702	0.594	0.493	0.300	0.545	0.728	0.575
	OMP (Ours)	**0.535±0.008**	0.809±0.015	0.831±0.007	**0.624±0.005**	**0.604±0.004**	**0.330±0.021**	**0.690±0.028**	**0.766±0.012**	**0.649±0.005**
Num=4	MC-SMoE	0.212	0.277	0.495	0.277	0.245	0.108	0.491	0.496	0.325
	HC-SMoE	0.322	0.613	0.754	0.493	0.392	0.256	0.614	0.671	0.514
	NAEE	**0.489**	**0.782**	0.814	0.577	0.473	0.290	0.614	0.729	0.596
	DiEP	0.473	0.768	0.812	**0.580**	0.489	0.292	0.599	**0.740**	0.594
	MoNE	0.310	0.562	0.620	0.412	0.231	0.202	0.527	0.647	0.439
	OMP (Ours)	0.482±0.009	0.781±0.015	**0.830±0.007**	0.577±0.005	**0.497±0.004**	**0.298±0.020**	**0.661±0.029**	0.728±0.012	**0.607±0.004**

GPT-OSS-20B, and 0.607 on Mixtral. These consistent task-level results show that the aggregate improvements in the main text are not driven by a single model family.

D.2 Comparison with One-Shot Expert Pruning Baselines

We further compare OMP-MoE with REAP Lasby et al. (2026) and EASY-EP Dong et al. (2025), two recent one-shot expert-pruning methods. All methods are evaluated on Qwen3-30B-A3B under the same checkpoint, pruning ratio, router treatment, evaluation harness, and 64-sequence WikiText-2 calibration protocol.

Table 19: Comparison with REAP and EASY-EP on Qwen3-30B-A3B. Bold indicates the best result among the three pruning methods at the same pruning ratio.

Pruned	Method	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.
0%	Original	0.528	0.792	0.887	0.596	0.778	0.346	0.823	0.703	0.682
25%	REAP	**0.555**	0.797	0.867	0.579	0.733	0.330	0.791	0.690	0.668
	EASY-EP	0.511	0.791	0.887	**0.593**	0.734	**0.344**	**0.809**	0.695	0.670
	OMP-MoE	0.534	**0.803**	**0.890**	0.592	**0.747**	0.338	0.805	**0.699**	**0.676**
50%	REAP	0.455	0.741	0.821	0.464	0.546	0.316	0.737	0.651	0.591
	EASY-EP	0.506	0.780	0.865	0.527	**0.613**	0.320	0.737	0.672	0.627
	OMP-MoE	**0.533**	**0.795**	**0.879**	**0.546**	0.605	**0.324**	**0.765**	**0.695**	**0.643**

The results are shown in the table 19. At 50% pruning, the margins increase to 0.052 over REAP and 0.016 over EASY-EP. Among these three recent pruning methods, OMP-MoE achieves the best score on seven of the eight tasks. The larger margin under stronger pruning indicates that residual-dependent expert selection is more useful when the retained expert budget is limited.

D.3 Compatibility with Orthogonal Compression Methods

We conduct an evaluation of the OMP-MoE integrated with post-training quantization and unstructured weight pruning algorithms. Table 20 reports the performance of perplexity and zero-shot accuracy on DeepSeek-V2-Lite model at a 25% pruning ratio. These results demonstrate that OMP MoE remains functional when combined with GPTQ Frantar et al. (2023) or SparseGPT Frantar & Alistarh (2023) or Wanda Sun et al. (2024). This modularity allows for multi level compression without catastrophic performance degradation, and is complementary to progressive low-bit quantization Xu et al. (2026).

Table 20: Performance comparison of OMP-MoE combined with post-training quantization and weight pruning methods on DeepSeek-V2-Lite at a 25% pruning ratio.

Methods	C4 ↓	WikiText-2↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg. ↑
OMP-MoE 25%	10.618	13.824	0.456	0.774	0.713	0.583	0.494	0.320	0.574	0.697	0.576
+ GPTQ	12.438	14.236	0.404	0.739	0.743	0.405	0.508	0.254	0.621	0.657	0.541
+ SparseGPT (4:8)	13.709	15.673	0.369	0.711	0.720	0.501	0.368	0.302	0.578	0.655	0.525
+ Wanda	13.321	14.430	0.394	0.728	0.720	0.501	0.390	0.308	0.585	0.649	0.534

D.4 Sensitivity Studies on Calibration Set Size

We analyze the impact of calibration data volume and domain on the pruning performance. Table 21 shows the results for Qwen3-30B-A3B at a 50% pruning ratio. The accuracy remains stable as the sample size increases from 16 to 256. We find that 64 samples provide sufficient information for near-optimal expert selection. The performance is consistent across both C4 Raffel et al. (2020) and WikiText-2 Merity et al. (2017) calibration sets.

Table 21: Impact of calibration sample size and dataset source on Qwen3-30B-A3B performance at a 50% pruning ratio.

Calibration Dataset	Number of Samples	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.↑
C4	16	0.428	0.708	0.867	0.575	0.544	0.312	0.758	0.702	0.612
	32	0.460	0.742	0.870	0.572	0.554	0.308	**0.791**	0.692	0.624
	64	0.457	0.737	0.872	**0.579**	0.574	0.332	0.765	**0.703**	0.627
	128	0.482	0.777	0.874	0.575	0.578	0.320	0.736	0.695	**0.630**
	256	**0.497**	**0.778**	**0.880**	0.575	**0.583**	**0.338**	0.675	0.702	0.628
WikiText-2	16	0.517	0.778	0.873	0.548	0.606	0.316	0.762	0.681	0.635
	32	0.526	0.781	**0.880**	0.550	0.617	**0.334**	0.751	0.689	0.641
	64	**0.533**	**0.795**	0.879	0.546	0.605	0.324	0.765	**0.695**	**0.643**
	128	0.521	0.792	0.878	0.549	0.614	0.314	0.729	0.692	0.636
	256	0.486	0.753	0.844	**0.564**	**0.698**	0.326	**0.769**	0.662	0.638

D.5 Ablation of Adaptive Inference (OMP-MoE†)

We evaluate the trade-off between inference efficiency and model fidelity using the OMP-MoE† mechanism. Table 22 summarizes the performance of DeepSeek-V2-Lite under different thresholds ϵ. We report the average expert activation count, expert skip ratios, and the perplexity on WikiText-2 and C4 datasets, and the zero-shot average accuracy scores on 8 benchmarks. The results indicate that a threshold ϵ of 0.1 balances FLOPs and accuracy effectively.

D.6 Empirical Inference Efficiency across Diverse Architectures

We present the empirical evaluation of inference performance and efficiency trade offs across four distinct Mixture of Experts architectures Zhu et al. (2026a). Tables 23, 24, 25, and 26 summarize the average zero-shot accuracy, the total computational cost measured in seconds, and the relative speedup for GPT-OSS-20B, Mixtral-8×7B, Qwen3-30B-A3B, and DeepSeek-V2-Lite respectively. Each evaluation compares the static expert pruning framework against the adaptive OMP-MoE† mechanism under the global budget allocation strategy. The results demonstrate that the synergy between structured pruning and dynamic expert activation yields substantial reductions in execution latency while preserving the linguistic reasoning proficiency of the model across various parameter scales.

Table 22: Impact of the OMP-MoE† threshold ϵ on the inference efficiency and accuracy of DeepSeek-V2-Lite at a 25% pruning ratio.

Threshold ϵ	**Average Experts**	**Skip Ratio**	**WikiText-2↓**	**C4 ↓**	**Average Accuracy ↑**
0 (Static)	6	0.00%	13.824	10.618	0.576
0.001	5.97	0.50%	13.180	10.405	0.575
0.01	5.73	4.50%	13.061	10.333	0.576
0.05	4.68	22.00%	13.062	10.341	0.578
0.1 (Default)	3.90	35.00%	13.576	10.522	0.581
0.2	2.90	51.70%	13.580	10.785	0.570
0.3	2.42	59.60%	14.205	11.170	0.554

Table 23: Inference performance and efficiency trade offs for the GPT-OSS-20B model.

Pruning Ratio	OMP-MoE	OMP-MoE†	Avg. Acc	Cost ↓	Speedup ↑
0	-	-	0.574	3188s	1.00×
0.25	✓	-	0.565	2140s	1.49×
0.25	✓	✓	0.562	2129s	1.50×
0.50	✓	-	0.515	1705s	1.87×
0.50	✓	✓	0.510	1596s	2.00×

Table 24: Inference performance and efficiency trade offs for the Mixtral-8×7B model.

Pruning Ratio	OMP-MoE	OMP-MoE†	Avg. Acc	Cost ↓	Speedup ↑
0	-	-	0.675	2954s	1.00×
0.25	✓	-	0.649	2685s	1.10×
0.25	✓	✓	0.644	2517s	1.17×
0.50	✓	-	0.607	2382s	1.24×
0.50	✓	✓	0.600	2315s	1.28×

Table 25: Inference performance and efficiency trade offs for the Qwen3-30B-A3B model.

Pruning Ratio	OMP-MoE	OMP-MoE†	Avg. Acc	Cost ↓	Speedup ↑
0	-	-	0.682	9980	1.00×
0.25	✓	-	0.676	8911s	1.12×
0.25	✓	✓	0.674	8382s	1.19×
0.50	✓	-	0.643	6698s	1.49×
0.50	✓	✓	0.641	6447s	1.55×

Table 26: Inference performance and efficiency trade offs for the DeepSeek-V2-Lite model.

Pruning Ratio	OMP-MoE	OMP-MoE†	Avg. Acc	Cost ↓	Speedup ↑
0	-	-	0.609	2459s	1.00×
0.25	✓	-	0.576	2222s	1.11×
0.25	✓	✓	0.581	2019s	1.22×
0.50	✓	-	0.489	1825s	1.35×
0.50	✓	✓	0.476	1690s	1.46×

Across Tables 23–26, adaptive execution provides additional latency reduction after static expert pruning. At 50% pruning, the combined speedups range from 1.28× on Mixtral to 2.00× on GPT-OSS-20B. The corresponding accuracy changes are small on GPT-OSS-20B, Mixtral, and Qwen3, while DeepSeek-V2-Lite exhibits a larger decrease from 0.489 to 0.476, illustrating that the adaptive threshold retains an architecture-dependent accuracy-efficiency trade-off.

D.7 Robustness and Scaling Results

We test robustness beyond the primary 25% and 50% pruning settings from three complementary perspectives: more aggressive expert removal, transfer to a substantially larger MoE model, and comparison with smaller dense alternatives. Table 27 evaluates Qwen3-30B-A3B when only 40% or 25% of its experts are retained. Table 28 evaluates scaling to Qwen3-235B-A22B, and Table 29 compares the 50%-pruned Qwen3-30B-A3B with dense Qwen3 models.

Table 27: Performance under stronger retained-expert constraints on Qwen3-30B-A3B. "Retained" denotes the fraction of experts kept in each MoE layer.

Retained	Method	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.
100%	Original	0.528	0.792	0.887	0.596	0.778	0.346	0.823	0.703	0.682
40%	NAEE	0.276	0.457	0.623	0.401	0.239	0.222	0.610	0.617	0.431
40%	DiEP	0.299	0.562	0.775	0.425	**0.375**	0.224	0.570	0.630	0.482
40%	OMP-MoE	**0.372**	**0.636**	**0.848**	**0.553**	0.235	**0.284**	**0.736**	**0.705**	**0.546**
25%	NAEE	0.195	0.283	0.419	0.262	**0.233**	0.132	**0.588**	0.507	0.327
25%	DiEP	0.212	0.423	0.622	0.316	0.229	0.142	0.545	0.556	0.381
25%	OMP-MoE	**0.241**	**0.453**	**0.712**	**0.427**	0.230	**0.192**	0.538	**0.636**	**0.429**

Under aggressive compression in Table 27, OMP-MoE achieves average accuracies of 0.546 and 0.429 when retaining 40% and 25% of experts, exceeding DiEP by 0.064 and 0.048 and NAEE by 0.115 and 0.102, respectively. Performance declines as expected when three quarters of the experts are removed, but the relative advantage of residual-conditioned selection remains.

Table 28: Performance on Qwen3-235B-A22B at a 50% pruning ratio. We report eight-task accuracy and search time.

Num	Method	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.	Time (s)
128	Original	0.618	0.854	0.896	0.678	0.849	0.372	0.812	0.787	0.733	–
64	NAEE	0.486	0.737	0.841	0.591	0.680	0.296	**0.747**	0.740	0.640	20491
64	DiEP	0.513	0.789	0.854	0.611	0.698	0.344	0.711	0.757	0.660	10473
64	OMP-MoE	**0.559**	**0.822**	**0.898**	**0.667**	**0.724**	**0.358**	0.682	**0.774**	**0.685**	**846**

The scaling results in Table 28 show that OMP-MoE reaches 0.685 average accuracy on Qwen3-235B-A22B, compared with 0.660 for DiEP and 0.640 for NAEE. Its 846-second search is more than 12× faster than DiEP and 24× faster than NAEE, indicating that the cached greedy search remains practical as the model size increases.

Table 29: Comparison between the pruned Qwen3-30B-A3B MoE model and smaller dense Qwen3 models across eight tasks.

Model	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Avg.
Qwen3-1.7B dense	0.399	0.725	0.777	0.461	0.524	0.278	0.708	0.615	0.561
Qwen3-4B dense	0.503	0.752	0.850	0.522	0.530	0.296	0.747	0.654	0.607
Qwen3-30B-A3B, 50% pruned	**0.533**	**0.795**	**0.879**	**0.546**	**0.605**	**0.324**	**0.765**	**0.695**	**0.643**

Table 29 provides a deployment-oriented reference point. The 50%-pruned Qwen3-30B-A3B obtains an average accuracy of 0.643, outperforming the dense Qwen3-4B and Qwen3-1.7B models by 0.036 and 0.082, respectively. Thus, structured expert removal retains an accuracy advantage over replacing the MoE with these smaller dense checkpoints.

D.8 Domain-Specific Calibration on Generative Tasks

We next test whether task-relevant calibration better preserves specialized experts. For GSM8K, we calibrate on `allenai/tulu-3-sft-personas-math`. For HumanEval, we use `theblackcat102/evol-codealpaca-v1`. Each training split is shuffled with seed 42 before constructing 64 sequences of 4,096 tokens. OMP-MoE, REAP, and EASY-EP use exactly the same calibration sequences and raw-token calibration format. We keep the OMP-MoE hyperparameters fixed from the eight-task development setting and do not retune them on GSM8K or HumanEval.

Table 30: Domain-specific calibration on Qwen3-30B-A3B. GSM8K uses 5-shot exact match with mathematical calibration, while HumanEval uses 0-shot pass@1 with code calibration and the `humaneval_instruct` evaluation protocol. Bold indicates the best pruned result at each pruning ratio.

Task	Method	Full (0%)	25%	50%
GSM8K	Full model	0.891	–	–
	OMP-MoE	–	**0.897**	**0.898**
	REAP	–	0.895	0.895
	EASY-EP	–	0.895	0.888
HumanEval	Full model	0.933	–	–
	OMP-MoE	–	**0.939**	**0.896**
	REAP	–	0.927	0.878
	EASY-EP	–	0.921	0.872

With mathematical calibration, OMP-MoE reaches 0.897 and 0.898 exact match on GSM8K at 25% and 50% pruning, respectively, compared with 0.891 for the full model. With code calibration, it reaches 0.939 and 0.896 pass@1 on HumanEval, compared with 0.933 for the full model. OMP-MoE also outperforms REAP and EASY-EP at both pruning ratios on both tasks. These results show that the retained expert set is calibration-distribution dependent: general-domain calibration can preserve substantial reasoning ability, while domain-specific calibration can further recover specialized mathematical and code-generation capabilities.

D.9 Reconstruction Fidelity and Downstream Accuracy

We further study the relation between reconstruction error and downstream accuracy on Qwen3-30B-A3B. We compute the total reconstruction error E as the sum of normalized residual rates across all MoE layers. Table 31 reports this error together with the corresponding average accuracy across eight tasks at five pruning ratios. We define normalized reconstruction fidelity as

Table 31: Reconstruction error and average downstream performance on Qwen3-30B-A3B.

Pruning Ratio	Reconstruction Error E	Average Accuracy A
0%	0.000	0.682
25%	0.193	0.676
50%	2.633	0.643
60%	5.388	0.546
75%	17.047	0.429

$$\phi = 1 - \frac{E}{17.047}, \tag{32}$$

where E is the total reconstruction error and $\phi \in [0, 1]$. A least-squares fit over the five operating points gives

$$A \approx 0.4170 + 0.2533\phi, \qquad R^2 \approx 0.947. \tag{33}$$

The positive fitted slope shows that higher reconstruction fidelity is associated with higher downstream average accuracy. This supports the use of reconstruction residual as the optimization target in OMP-MoE, while the sensitivity analysis in Appendix A.5 explains why local deviations may occur when residuals lie in low-sensitivity directions.

E VISUALIZATION AND ANALYTICAL EXPERIMENTS

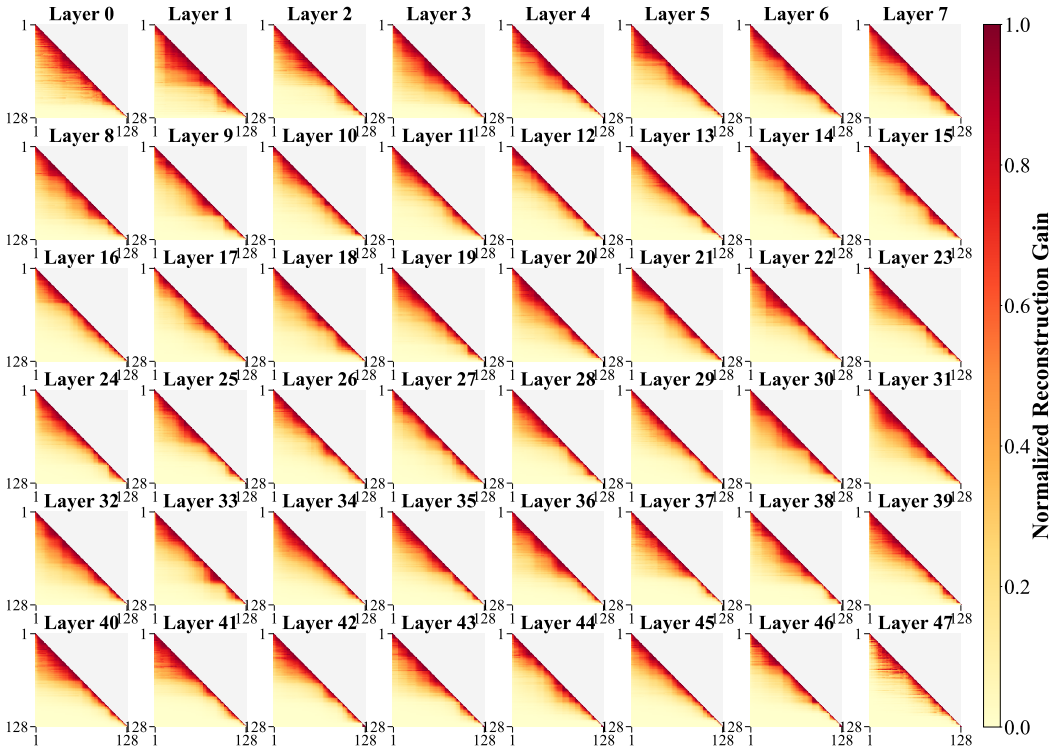
We provide a comprehensive visual analysis of the OMP-MoE process. This section includes layer-wise results for four different MoE architectures. These models are Qwen3-30B-A3B, DeepSeek-V2-Lite, GPT-OSS-20B, and Mixtral-8$\times$7B.

E.1 RECONSTRUCTION GAIN HEATMAPS

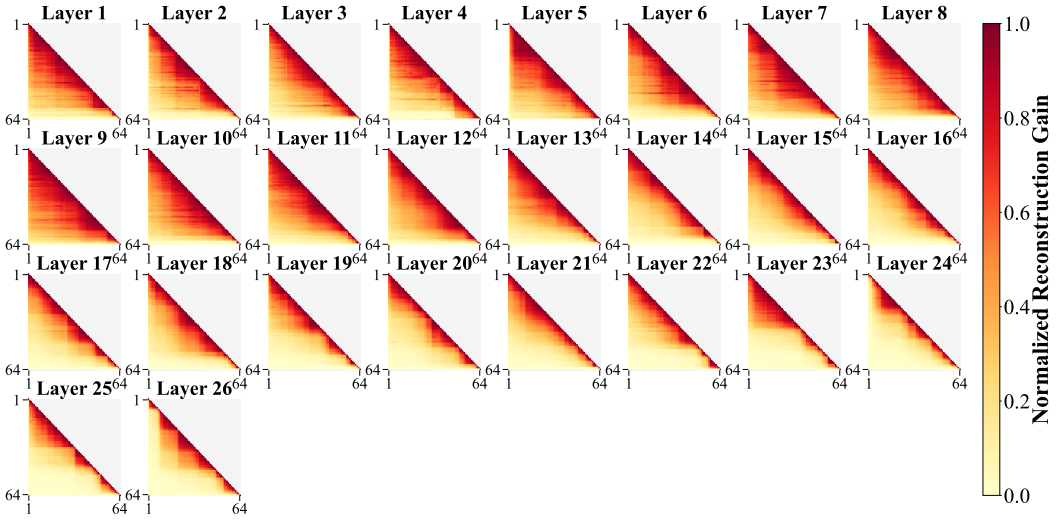
We visualize the dynamic importance of experts through reconstruction gain heatmaps. These heatmaps illustrate the relative contribution of each candidate expert at every selection step. A higher color intensity represents a larger reduction in the residual signal. Figure 4 and 5 present the layer-wise heatmap results for all four experimental models. These visualizations reveal how the priority of experts shifts during the iterative search process. The OMP algorithm effectively disentangles the correlations between experts by updating the residual signal.

We visualize the reconstruction gain heatmaps to illustrate the dynamic selection of experts throughout the greedy selection process. Each cell within the grid represents the reconstruction gain Δ_e for a candidate expert at a specific selection iteration. The horizontal axis tracks the progression of selection steps while the vertical axis denotes individual expert indices within the MoE layer. The varying color intensity characterizes the marginal contribution of an expert to the current residual signal. Bright regions indicate experts with high correlation to the uncaptured information manifold, whereas darker regions signify high redundancy relative to the previously selected subset.

These heatmaps reveal the conditional nature of expert importance because the intensity patterns shift significantly after each iteration step. We observe that experts exhibit mutual dependencies during the iterative pursuit. The selection of a specific expert alters the residual signal energy. And this modification in the residual energy subsequently modifies the reconstruction gain of all remaining expert candidates. The pattern of mutual dependencies in experts confirms that one-shot ranking methodologies are insufficient for identifying optimal expert subsets. Furthermore, the patterns highlight that specific experts consistently dominate the initial reconstruction phase in shallow layers, whereas deeper layers exhibit more distributed importance across the expert dictionary. We provide the full layer wise heatmap results for the four primary experimental architectures in Figures 4 and 5.

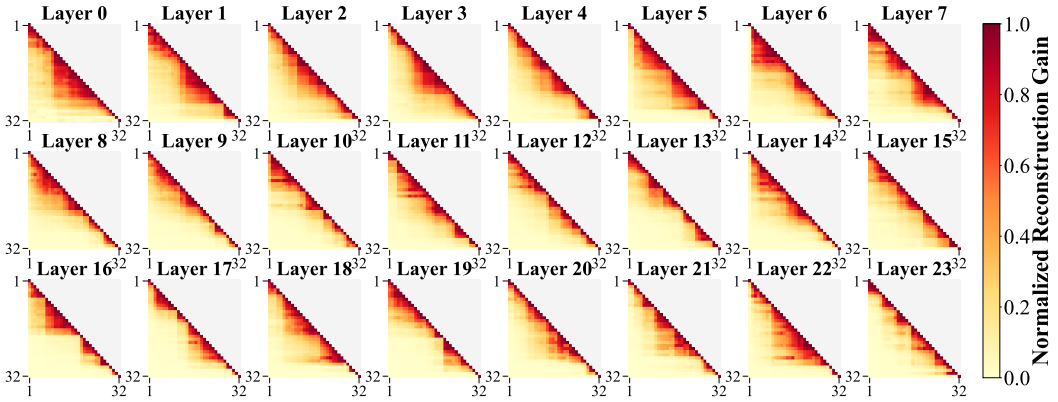


(a) Visualization of Layer-wise reconstruction gain heatmaps within each MoE layer on Qwen3-30B-A3B.

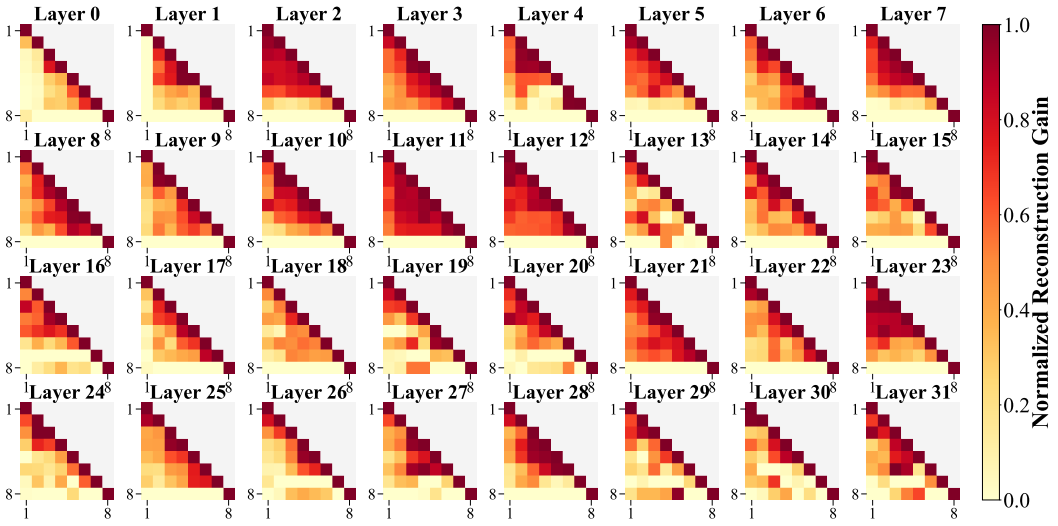


(b) Visualization of Layer-wise reconstruction gain heatmaps within each MoE layer on DeepSeek-V2-Lite.

Figure 4: Layer-wise reconstruction gain heatmaps for the Qwen3-30B-A3B and DeepSeek-V2-Lite models. Each row represents a specific MoE layer and each column represents a selection iteration. Bright cells indicate experts that provide the maximum reconstruction gain at that step.



(a) Visualization of Layer-wise reconstruction gain heatmaps within each MoE layer on GPT-OSS-20B.



(b) Visualization of Layer-wise reconstruction gain heatmaps within each MoE layer on Mixtral-8×7B.

Figure 5: Layer-wise reconstruction gain heatmaps for the GPT-OSS-20B and Mixtral-8×7B models. Each row represents a specific MoE layer and each column represents a selection iteration. Bright cells indicate experts that provide the maximum reconstruction gain at that step.

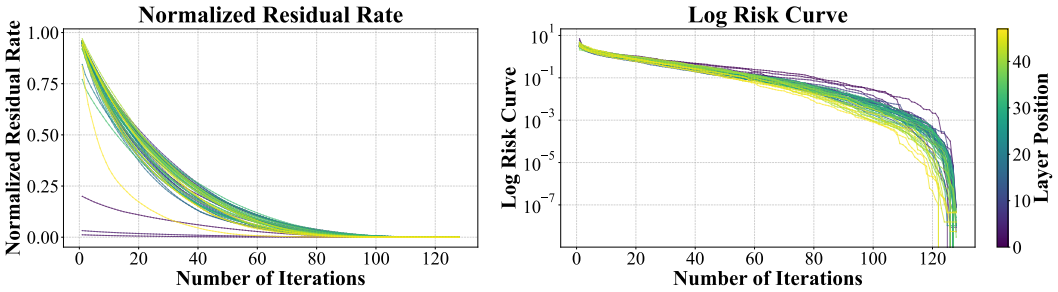
E.2 Normalized Residual Curves

We investigate the convergence properties of the iterative selection process by examining the normalized residual curves across various MoE architectures. The horizontal axis quantifies the count of selected experts while the vertical axis represents the normalized Frobenius norm of the residual signal. Each curve characterizes the proportion of the original signal energy that remains uncaptured after a specific number of selection iterations. We observe that these trajectories typically exhibit a monotonic decay which signifies the progressive reconstruction of the layer output through the Orthogonal Matching Pursuit algorithm.

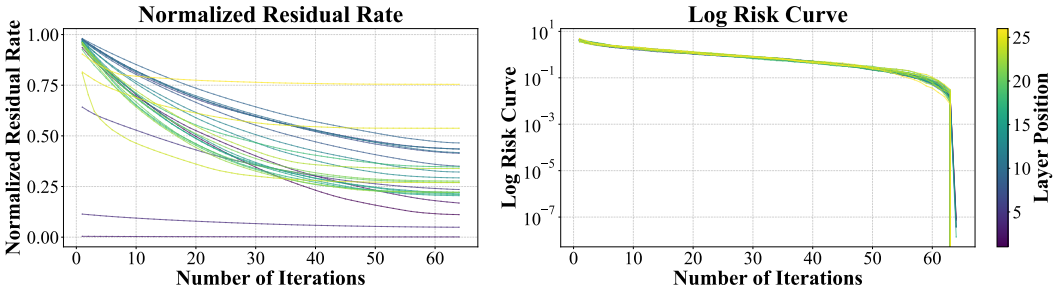
The gradient of the residual curve provides a direct metric for expert redundancy within a specific MoE layer. A rapid descent indicates that a small subset of experts encapsulates the majority of the information whereas a linear or slow decay suggests that the contribution energy is distributed across a wider dictionary of atoms. Shallow layers frequently demonstrate high signal concentration which allows for aggressive pruning without substantial reconstruction loss. Conversely, deeper layers often require a larger expert budget to satisfy the same reconstruction threshold.

Furthermore, we visualize the risk rate curves to illustrate the stability of the routing distribution as described in Section 3.3. These curves track the preserved routing mass as the expert count increases. The intersection of the residual and risk trajectories highlights the physical trade off between reconstruction fidelity and routing renormalization stability. This visualized heterogeneity across layers justifies our use of a discrete water filling strategy for global budget allocation because it prioritizes layers with the steepest gain to risk ratios. Figures 6 and 7 provide the complete layer wise results for all experimental models.

Across Figures 6 and 7, the decay rates differ substantially across both layers and architectures. This variation supplies the layer-specific marginal-gain curves used by the global allocator rather than assuming uniform redundancy throughout a model.

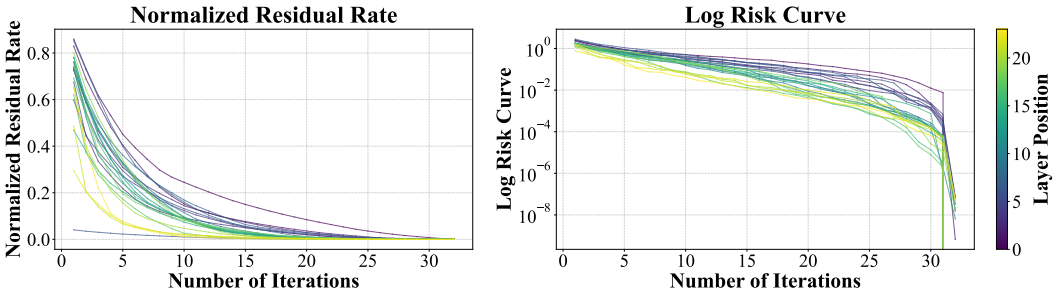


(a) Visualization of normalized residual and risk curves on the Qwen3-30B-A3B model.

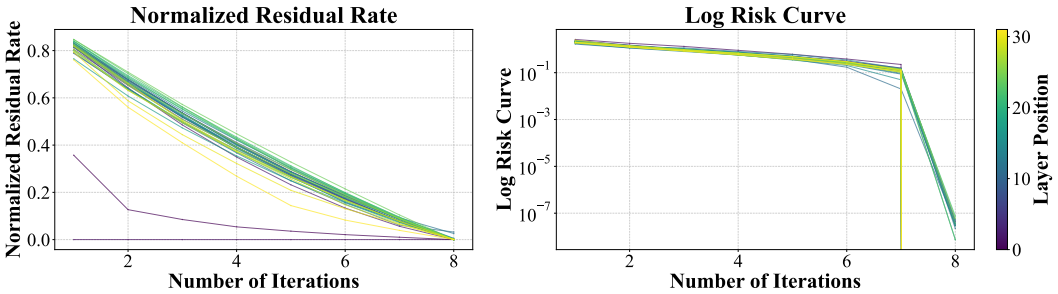


(b) Visualization of normalized residual and risk curves on the DeepSeek-V2-Lite model.

Figure 6: Normalized residual and risk curves for all layers on Qwen3-30B-A3B and DeepSeek-V2-Lite models. The horizontal axis represents the number of retained experts. The vertical axis represents the normalized residual rate and risk rate. Curves with rapid decay indicate layers with high expert redundancy.



(a) Visualization of normalized residual and risk curves on the GPT-OSS-20B model.



(b) Visualization of normalized residual and risk curves on the Mixtral-8×7B model.

Figure 7: Normalized residual and risk curves for all layers on GPT-OSS-20B and Mixtral-8×7B models. The horizontal axis represents the number of retained experts. The vertical axis represents the normalized residual rate and risk rate. Curves with rapid decay indicate layers with high expert redundancy.

E.3 Identification of High-Contribution Experts

We use the descriptive term *high-contribution expert* for an expert whose first-step OMP selection reconstructs more than 90% of a layer's routed-output energy. Operationally, this occurs when the normalized residual after the first greedy step satisfies $r_l(1) < 0.1$, corresponding to a first-step reconstruction gain greater than 0.9. Our definition is specific to the calibration-conditioned OMP objective and quantifies importance through residual reduction relative to the other candidate atoms in the same layer.

This phenomenon is distinct from the *Super Experts* introduced by Su et al. Su et al. (2025). That work identifies rare experts through extreme `down_proj` activation outliers and studies their data-agnostic role in hidden-state outliers, attention sinks, and downstream performance. In contrast, our analysis neither assigns that established name nor assumes the same activation mechanism: it uses the binary matching-pursuit view to measure how much routed-output energy a candidate uniquely explains under a particular calibration distribution. The two criteria may identify overlapping experts, but they answer different questions and are not treated as equivalent here.

Table 32: High-contribution experts across four MoE architectures, identified by the initial OMP reconstruction residual.

Model	Layer Index	Residual $r_l(1)$	Expert ID
DeepSeek-V2-Lite	3	0.0041	54
GPT-OSS-20B	6	0.0407	5
Mixtral-8×7B	1	0.0001	3
Qwen3-30B-A3B	2	0.0118	92
Qwen3-30B-A3B	3	0.0322	82

Table 32 identifies five representative cases across all four architectures. The first-step residual ranges from 0.0001 for Mixtral layer 1 to 0.0407 for GPT-OSS layer 6, so each listed expert reconstructs more than 95% of its layer's calibration output under this criterion. These cases provide empirical evidence of strong signal concentration: after the dominant atom is selected, the residual-conditioned marginal utility of subsequent experts diminishes rapidly. This OMP-based quantification also motivates the cross-layer allocator, which can assign smaller budgets to layers with rapidly decaying residual curves.

F Extended Discussion

In this section, we provide clarifications regarding the theoretical boundaries and implementation details of the OMP-MoE framework.

F.1 Scope of Greedy Optimality

We explicitly distinguish our selection mechanism from standard Orthogonal Matching Pursuit. The classical algorithm continuously updates coefficients via orthogonal projection. In contrast, expert pruning imposes a binary constraint where coefficients are fixed at one. Consequently, our optimality guarantee is scoped to the local greedy step. OMP-MoE ensures that each iteration selects the expert providing the maximum immediate reduction in residual energy conditioned on the current subset. This formulation avoids matrix inversion overhead while effectively handling mutual dependencies between experts.

F.2 Superiority over Static Heuristics

OMP-MoE addresses the inter-expert correlations that static ranking heuristics neglect. Methods based on one-shot routing frequency or norm assume expert independence. However, our heatmaps in Appendix E.1 demonstrate that the utility of a candidate expert shifts dynamically after selecting a dominant expert. By updating the residual signal, OMP-MoE re-evaluates candidates based on uncaptured information. This allows the identification of complementary experts that static magnitude-based methods would discard.

F.3 BROADER IMPACT

OMP-MoE aims to reduce the memory and inference cost of MoE LLMs, which may lower the hardware barrier for running large models and reduce energy use during deployment. This can make efficient language technologies more accessible to researchers and users with limited compute resources.