MoE Pathfinder: Trajectory-driven Expert Pruning

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-experts (MoE) architectures used in large language models (LLMs) achieve state-of-the-art performance across diverse tasks yet face practical challenges such as deployment complexity and low activation efficiency. Expert pruning has thus emerged as a promising solution to reduce computational overhead and simplify the deployment of MoE models. However, existing expert pruning approaches conventionally rely on local importance metrics and often apply uniform layer-wise pruning, leveraging only partial evaluation signals and overlooking the highly heterogeneous contributions of experts across layers. To address these limitations, we propose an expert pruning approach based on the trajectory of activated experts across layers, which treats MoE as a weighted computation graph and casts expert selection as a global optimal path planning problem. Within this framework, we integrate complementary importance signals from reconstruction error, routing probabilities, and activation strength at the trajectory level, which naturally yields non-uniform expert retention across layers. Experiments show that our approach achieves superior pruning performance on nearly all tasks compared with most existing approaches.

1 Introduction

Large language models (LLMs) almost dominate all tasks across a wide range of scenarios with strong performance (Brown et al., 2020; Wei et al., 2022; Liu et al., 2023; Li et al., 2023b; Lin et al., 2023; Touvron et al., 2023; Achiam et al., 2023). The recent progress on LLMs shows that the mixture-of-experts (MoE) architecture has emerged as a key paradigm because of its superior performance-compute trade-off (Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024; Liu et al., 2024a; Meta, 2025; Team et al., 2025a; Zeng et al., 2025; Team et al., 2026; Li et al., 2026a), where a sparse activation mechanism is utilized with a small subset of experts activated for each input, enabling large parameter scales without proportional computational overhead (Lepikhin et al., 2020; Artetxe et al., 2021). Owing to computational sparsity, MoE models require loading all expert weights during inference, no matter whether an expert is activated, resulting in a substantial random access memory (RAM) and storage burden[1]. Furthermore, in task-specific or domain-adapted scenarios, only a small subset of experts is usually enough to achieve strong performance, while activating all experts yields marginal gains but greatly reduces inference efficiency (Su et al., 2025; Lo et al., 2025). Therefore, to preserve the specialization of MoE models while improving inference efficiency, it is important to adopt expert pruning strategies that effectively operate on the MoE architecture.

Existing studies on compressing MoE architectures include dynamic expert selection, expert merging, and expert pruning. Dynamic expert selection introduces zero-computation experts or estimates token-level importance to dynamically adjust the number of activated experts during inference, but it does not reduce parameter counts (Jin et al., 2024; Raposo et al., 2024; Zeng et al., 2024; Yue et al., 2024; Team et al., 2025b). Expert merging reduces the number of experts by linearly combining weights, but it often harms expert specialization (Li et al., 2023a; He et al., 2023; Chen et al., 2024; Li et al., 2025). Expert pruning removes some experts and becomes the mainstream direction because it lowers memory and computational costs while effectively preserving expert specialization (Guo et al., 2025b; Zhang et al., 2025; Yoon & Choi, 2026; Hu et al., 2026). Existing expert pruning

[1] E.g., Mixtral-8x7B (Jiang et al., 2024) activates only a few experts per pass but must keep all parameters resident in RAM.

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

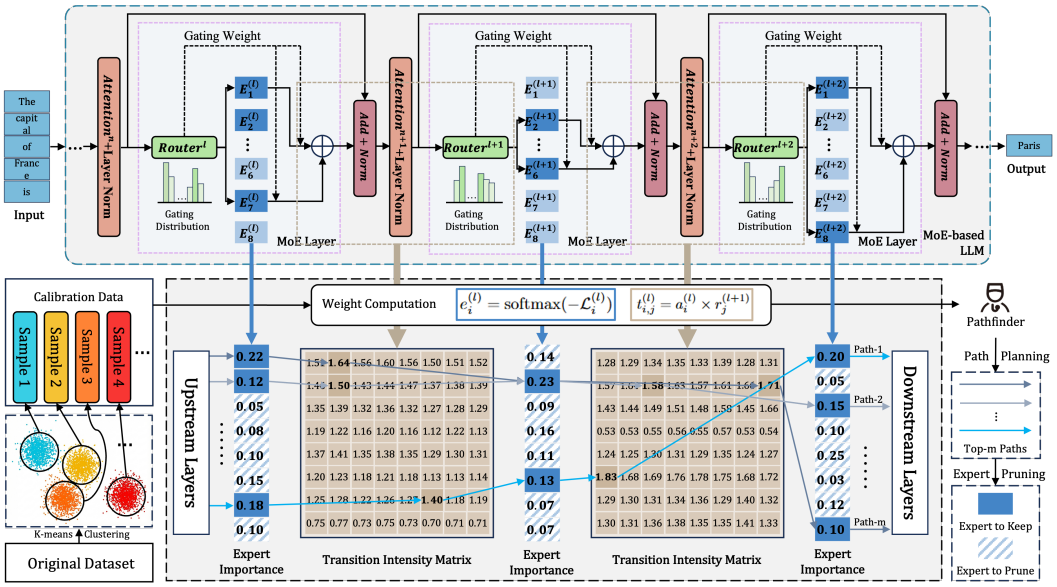

Figure 1: Overview of the proposed expert pruning framework based on the trajectory of activated experts across layers. The top part presents the standard multi-layer MoE architecture, where tokens are dynamically routed to sparse experts. The bottom part shows our pruning approach that reformulates the MoE as a directed weighted graph. In our approach, we firstly compute transition intensities (edge weights) and expert importance scores (node weights) based on routing probabilities, expert activations, and reconstruction loss, respectively. Then, we use a global path planning algorithm to identify the top-ranked optimal inference trajectories. Experts located on these critical paths are retained (highlighted in purple), while the remaining redundant experts are pruned.

approaches are broadly categorized into activation-based strategies and perturbation-based strategies. Activation-based strategies typically estimate expert importance using routing probabilities and expert activations (Muzio et al., 2024; Dong et al., 2025; Lasby et al., 2025). While computationally efficient, these approaches rely on local statistics and fail to accurately capture the experts' contribution to global task performance. In contrast, perturbation-based approaches assess importance by measuring expert output norms or performance discrepancies when experts are removed (Cao et al., 2024; Lu et al., 2024). These approaches offer a more comprehensive view of expert influence but require multiple forward passes or reconstruction-based sensitivity analysis, which incurs prohibitively high computational costs. Moreover, most existing approaches (Zhang et al., 2025; Lee et al., 2025) apply a uniform pruning ratio across all layers, ignoring the inherent heterogeneity of expert importance among layers, leading to insufficient resources for critical layers and insufficient compression for redundant ones, ultimately degrading overall performance and efficiency. Therefore, existing expert pruning approaches either depend on local heuristics or require expensive computations, making it difficult to achieve a globally consistent yet efficient evaluation of expert importance. As a result, there is a pressing need for a pruning strategy that integrates multi-dimensional information and accounts for cross-layer heterogeneity in a computationally efficient way. Intuitively, the routing mechanism in MoE inherently performs a path selection across layers, which naturally motivates reformulating expert selection as a global optimization problem on a graph.

In this paper, we propose an expert pruning approach that leverages the activation trajectories of experts across layers, which treats MoE as a weighted computation graph and casts expert selection as a global optimal path planning problem. We firstly model the MoE architecture as a layered graph, where experts act as nodes and inter-layer interactions are captured through transition intensity matrices. Within this graph, we quantify information flow using two complementary metrics: transition intensities and expert importance scores. Leveraging these metrics, we cast expert selection as identifying the top globally significant computation paths through a dynamic-programming-based path exploration scheme. By aggregating critical experts appearing in these high-value trajectories across calibration samples, we construct a unified pruning mask to realize structured MoE compression. This graph-driven global evaluation framework integrates activation

strength, routing behaviors, and reconstruction-based expert importance, enabling adaptive cross-layer non-uniform pruning while preserving the model's specialized capabilities. Experiment results show that our approach substantially reduces inference cost and memory usage while maintaining or surpassing baseline performance on benchmark datasets.

2 THE APPROACH

We propose an expert pruning approach based on the trajectory of activated experts across layers that performs pruning for task-specific MoE models. The workflow is illustrated in Figure 1, where the upper part outlines the standard MoE architecture and the lower part visualizes how the trajectory-driven approach is constructed and used for pruning via global path planning. In contrast to conventional pruning schemes that assess experts independently and often in a layer-wise manner, our approach reformulates the MoE computation as a structured graph. On this graph, we compute data-driven weights that capture both the intrinsic contribution of each expert and the cross-layer information flow between experts. We then perform global path planning to identify the paths most critical to the target task and prune away experts that do not participate in any of these high-importance paths. In the following sub-sections, we present the details of the graph construction, the computation of trajectory weights, and the global-path-planning strategy for expert pruning.

2.1 GRAPH CONSTRUCTION

We formalize the MoE architecture as a layer-wise graph that captures all feasible routes of information flow during inference. The purpose of this representation is to make expert-level dependencies across layers visible so that pruning decisions are made at the global path level rather than in an isolated, per-layer manner. Formally, the MoE model contains L expert layers. At layer l, we denote its N_e experts by the set $E_1^{(l)}, \ldots, E_{N_e}^{(l)}$. Each expert corresponds to a node in the trajectory. Stacking all layers together yields a directed acyclic graph composed of L groups of expert nodes, where edges only appear between adjacent layers. A complete inference trajectory is thus visualized as selecting one expert per layer, connected through the transition intensities across successive layers. This graph abstraction allows the information-propagation pattern to be explicit and establishes a unified foundation for the subsequent weight computation and global path–based expert pruning.

2.2 WEIGHT COMPUTATION

Once the graph is constructed, we assign two weight metrics: the transition intensity between adjacent experts and the importance score of each expert. Transition intensity models the strength of cross-layer information flow, while expert importance reflects each expert's individual contribution. Together, these metrics define the quality of computation paths and allow pruning to consider both local expert quality and global path structure. All weights are computed from a calibration dataset.

Before describing the construction of these weights, we introduce notations for clarity. At layer l, the MoE module contains N_e experts with weight matrices $\mathbf{W}_1^{(l)}, \ldots, \mathbf{W}_{N_e}^{(l)}$. Each expert weight matrix has a shape of $(d_{\text{out}}, d_{\text{in}})$, where d_{in} denotes the input dimensionality of experts and d_{out} denotes the output dimensionality. The router at layer l is parameterized by a matrix $\mathbf{R}^{(l)} \in \mathbb{R}^{N_e \times d_{\text{in}}}$, which maps the layer input representation into routing logits over the N_e experts. Given a sample X from the calibration dataset D, we denote the hidden state entering layer l as $\mathbf{H}^{(l-1)} \in \mathbb{R}^{N_x \times d_{\text{in}}}$, where N_x is the number of tokens in the sample. For the first layer, we have $\mathbf{H}^{(0)} = \mathbf{X}$. In the following text, we detail how the transition intensities and expert importance scores are computed.

Transition Intensity To encode the flow of computations across layers, we introduce a fully connected transition intensity matrix $\mathbf{T}^{(l)} = \left(t_{i,j}^{(l)}\right)_{N_e \times N_e}$ between layer l and layer $l+1$, where each entry $t_{i,j}^{(l)}$ represents the transition strength from expert $E_i^{(l)}$ to expert $E_j^{(l+1)}$. To make this quantity more interpretable, we decompose the transition into two components, namely, the upstream activation strength $a_i^{(l)}$ and the downstream routing probability $r_j^{(l+1)}$.

Specifically, for the upstream activation strength, it quantifies how strongly expert i effectively contributes to token representations at layer l. For the k-th token, the corresponding output magnitude

of expert i is computed as

$$a_{i,k}^{(l)} = \left\| \mathbf{H}_k^{(l-1)} (\mathbf{W}_i^{(l)})^\top \right\|_2 \tag{1}$$

where $\|\cdot\|_2$ computes the ℓ_2-norm of a vector. Averaging over all tokens yields the upstream activation strength:

$$a_i^{(l)} = \frac{1}{N_x} \sum_{k=1}^{N_x} a_{i,k}^{(l)} \tag{2}$$

This term captures the amount of signal emitted by expert i.

For downstream routing probability, we measure how likely expert j is to be selected by the router in layer $l+1$. For each token, its routing probability $r_{j,k}^{(l+1)}$ is computed as

$$r_{j,k}^{(l+1)} = \left[\mathrm{softmax}\left(\mathbf{H}_k^{(l)} (\mathbf{R}^{(l+1)})^\top \right) \right]_j \tag{3}$$

Averaging tokens gives the downstream routing preference:

$$r_j^{(l+1)} = \frac{1}{N_x} \sum_{k=1}^{N_x} r_{j,k}^{(l+1)} \tag{4}$$

This term captures the likelihood that expert j will receive information in the next layer.

Combining upstream activation strength and downstream routing probability, we obtain overall transition intensity:

$$t_{i,j}^{(l)} = a_i^{(l)} \times r_j^{(l+1)} \tag{5}$$

This factorization highlights that a strong transition occurs only when the upstream expert emits a large activation and the downstream expert has a high routing probability.

Expert Importance For expert i in layer l, we define an expert importance score $e_i^{(l)}$ that quantifies its individual contribution to the layer's output. To obtain this score, we measure how well expert i alone is able to faithfully reconstruct the original output of layer l.

Firstly, we denote the original output of layer l for token k as $\mathbf{Y}_k^{(l)}$. When expert i is active, its reconstructed output is

$$\hat{\mathbf{Y}}_{k,i}^{(l)} = \mathbf{H}_k^{(l-1)} (\mathbf{W}_i^{(l)})^\top \tag{6}$$

This provides a token-wise view of how much expert i alone contributes to the layer computation. Next, the discrepancy between the original output and the expert-only reconstruction is quantified by the per-expert reconstruction loss:

$$\mathcal{L}_i^{(l)} = \frac{1}{N_x} \sum_{k=1}^{N_x} \left\| \mathbf{Y}_k^{(l)} - \hat{\mathbf{Y}}_{k,i}^{(l)} \right\|_2^2 \tag{7}$$

A smaller loss indicates that expert i more faithfully captures the behavior of the full MoE layer. Then, we convert reconstruction losses into importance scores via a softmax:

$$e_i^{(l)} = \mathrm{softmax}(-\mathcal{L}_i^{(l)}) \tag{8}$$

assigning greater importance to experts who more closely approximate the original output.

Different from intermediate layers, the first and last layers lack upstream or downstream routing signals, respectively. To capture these missing directional flows, we incorporate additional correction terms. For the first layer:

$$e_i^{(1)} = \mathrm{softmax}(-\mathcal{L}_i^{(1)}) \times \frac{1}{N_x} \sum_{k=1}^{N_x} \left[\mathrm{softmax}\big(\mathbf{X}_k (\mathbf{R}^{(1)})^\top \big) \right]_i \tag{9}$$

where the notation $[\cdot]_i$ extracts the i-th entry of the softmax vector, indicating the routing probability that the router of the first layer assigns to expert i. For the last layer:

$$e_i^{(L)} = \mathrm{softmax}(-\mathcal{L}_i^{(L)}) \times \frac{1}{N_x} \sum_{k=1}^{N_x} \left\| \mathbf{H}_k^{(L-1)} (\mathbf{W}_i^{(L)})^\top \right\|_2 \tag{10}$$

which incorporates the activation strength produced by expert i. These adjustments ensure that the importance scores for boundary layers reflect the same flow semantics as intermediate layers.

2.3 PRUNING VIA GLOBAL PATH PLANNING

After constructing the weighted graph, we identify essential experts by analyzing the computation paths that span from the first MoE layer to the last. The key idea is that each path represents a potential sequence of expert activations during inference. We therefore search for the top-m most important paths and retain only the experts that appear along these paths. To enable this selection, we assign a weight to each feasible path based on the expert- and transition-level quantities obtained from Sec. 2.2. Specifically, each path weight integrates two components: the transition intensities $t_{i,j}^{(l)}$, which capture the strength of information flow from layer l to $l{+}1$, and the expert importance scores $e_i^{(l)}$, which reflect the contribution of each expert within its layer. Together, these quantities provide a unified assessment of how effectively a path propagates information through the network.

Let $\mathcal{P}$ denote the feasible paths that select one expert per layer. For any path $p \in \mathcal{P}$, its weight w_p is defined as $w_p = \prod_{l=1}^{L-1} t_{i_l,i_{l+1}}^{(l)} \prod_{l=1}^{L} e_{i_l}^{(l)}$, where i_l denotes the expert selected by path p at layer l. This multiplicative structure reflects that a path is important only when it involves influential experts and strong transitions across layers. To avoid numerical instability from multiplying small values, the path weight is evaluated in logarithmic domain, yielding the following additive formulation:

$$\log w_p = \sum_{l=1}^{L-1} \log t_{i_l,i_{l+1}}^{(l)} + \sum_{l=1}^{L} \log e_{i_l}^{(l)} \tag{11}$$

The top-m path selection problem for MoE pruning is equivalent to identifying the m paths with the highest log-weights. Since every feasible path selects exactly one expert per layer, all paths have identical length. This ensures that the multiplicative weighting does not favor shorter paths, and differences in path weights arise solely from the quality of the experts and transitions along the path.

Since enumerating all feasible paths is infeasible, we adopt dynamic programming to constructs the highest-weight partial paths. We define a prefix path as a partial path spanning layers 1 through l, represented by the sequence of expert indices selected up to layer l. For each expert node v in layer l, we maintain a priority queue Q_v containing up to m prefix paths that terminate at v, each stored along with its log-weight. Then, we process nodes layer by layer in topological order. Specifically, for every incoming edge from expert i_l in layer l to expert i_{l+1} in layer $l{+}1$, each prefix path in Q_{i_l} is extended by appending expert i_{l+1}. The log-weight of the extended prefix path is computed as

$$\log w_{\text{new}} = \log w_{\text{old}} + \log t_{i_l,i_{l+1}}^{(l)} + \log e_{i_{l+1}}^{(l+1)} \tag{12}$$

where $\log w_{\text{old}}$ denotes the log-weight of the prefix path before extension. For each node, only the m extended paths with the highest weights are retained in its priority queue.

This dynamic programming procedure is applied independently to each sample in the calibration dataset, producing a set of top-m full paths per sample. For a given sample X, let $\mathcal{P}_{m,X}$ denote its corresponding selected top-m paths explicitly. The experts preserved for the sample X are defined as the union of all experts appearing in these respective paths $\mathcal{E}_{\text{keep},X} = \bigcup_{p \in \mathcal{P}_{m,X}} \mathcal{V}(p)$. At the dataset level, the final set of experts retained by the pruning procedure is the union of per-sample expert sets $\mathcal{E}_{\text{keep}} = \bigcup_{X \in \mathcal{D}} \mathcal{E}_{\text{keep},X}$. To apply pruning to the MoE model, a binary mask is constructed for each layer to disable experts not included in $\mathcal{E}_{\text{keep}}$:

$$M_i^l = \begin{cases} 1, & E_i^{(l)} \in \mathcal{E}_{\text{keep}} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

After determining the mask, the model is pruned by removing the parameters of unselected experts and by retaining the rows in the routing matrix that correspond to surviving experts. This pruning procedure significantly reduces both the memory footprint and computational cost during inference while preserving the most critical expert pathways.

3 EXPERIMENT SETTINGS

3.1 DATASETS

We evaluate on six widely used benchmarks: MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC (Clark et al., 2018), GSM8K (Cobbe

Table 1: Overall performance comparison of our expert pruning approach against baselines. This table compares the performance of "Full" (unpruned), "Random" (random baseline), and "Ours" (our approach) for Mixtral-8x7B and Mixtral-8x7B-Instruct models. All pruning approaches are set to 50% expert sparsity. All reported metrics are accuracy scores, where higher is better. "Avg." refers to the average accuracy across MMLU, HellaSwag, WinoGrande, ARC, GSM8K, and MedQA.

(a) Mixtral-8x7B

Approach	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
Full	67.88	64.83	76.08	56.48	58.45	62.37	64.35
Random	25.39	40.62	56.98	23.03	1.13	26.08	28.87
Ours	54.31	56.51	74.19	49.66	38.67	49.25	53.77

(b) Mixtral-8x7B-Instruct

Approach	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
Full	68.81	67.61	76.37	62.54	65.04	60.88	66.88
Random	31.12	46.76	56.83	29.61	1.90	30.87	34.52
Ours	57.66	60.42	73.01	53.18	30.10	46.74	53.52

et al., 2021), and MedQA (Jin et al., 2021), covering general knowledge, commonsense reasoning, science, mathematics, and medical QA. For each task, we construct a task-specific calibration set from its training split using k-means clustering. Following Dong et al. (2025), we replace the desired output of each calibration sample with the output of an LLM. Dataset statistics, full descriptions, and calibration details are provided in Appendix B.1.

3.2 Implementation Details

We conduct experiments on two widely adopted MoE-based LLMs, namely, Mixtral-8x7B and Mixtral-8x7B-Instruct (Jiang et al., 2024).[2] Mixtral-8x7B is the general pre-trained base model designed for broad language understanding, while Mixtral-8x7B-Instruct is its instruction-tuned variant optimized for responding to user prompts and following instructions. Both Mixtral-8x7B and Mixtral-8x7B-Instruct consist of 32 Transformer layers with a hidden dimension of 4,096, and each layer incorporates an MoE block containing eight experts from which two are activated per token. For evaluation, we use the EleutherAI LM Harness[3] to assess the accuracy of different models.

For different tasks, we use different cluster numbers K to partition the calibration dataset into clusters. Specifically, we set $K=12$ for MMLU and WinoGrande, $K=20$ for HellaSwag, $K=10$ for ARC, $K=9$ for GSM8K, and $K=5$ for MedQA.[4] To precisely control the expert coverage across varying models and compression ratios, which is crucial for subsequent pruning, we regulate the path planning process via a top-m parameter. For the Mixtral series, we set $m=1$ for pruning with an expert sparsity of 50%, and increase it to 500 when performing pruning under a 25% sparsity level.

4 Results and Analysis

4.1 Overall Results

We compare our expert pruning approach against the original full model and random expert pruning baseline across two distinct LLMs (i.e., Mixtral-8x7B and Mixtral-8x7B-Instruct). We summarize the main results in Table 1, where several observations are drawn from the results. First, our approach outperforms the random pruning baseline across all models. Furthermore, it approaches the performance of the full model on certain tasks, such as WinoGrande. Even on the strictly evaluated GSM8K benchmark, our approach maintains a degree of performance preservation. This phenomenon strongly proves that our approach is not merely removing experts, but possesses the

[2]The LLMs are obtained from their official HuggingFace repositories `https://huggingface.co/`.

[3]`https://github.com/EleutherAI/lm-evaluation-harness`

[4]We try different values of K and use the one that achieves the best performance for each dataset and provide a sensitivity analysis on the number of clusters K in Appendix A.6.

Table 2: Performance comparison of our pruning approach against existing MoE compression approaches for the Mixtral-8x7B model, with all approaches evaluated at 50% expert sparsity. "*" indicates that results obtained through replication, and was not taken directly from the original paper.

Technique	Approach	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
Merging	Li et al. (2023a)*	23.00	26.74	51.14	19.62	0.00	28.12	24.77
	Chen et al. (2024)	48.95	**57.81**	72.06	46.42	-	-	-
	Li et al. (2025)	48.00	57.00	72.00	45.00	-	-	-
Pruning	He et al. (2024)	25.54	42.50	56.99	22.35	-	-	-
	Lu et al. (2024)	47.30	57.66	72.85	48.89	-	-	-
	Muzio et al. (2024)	49.64	-	-	-	-	-	-
	Lee et al. (2025)*	24.19	25.83	50.51	23.12	0.00	24.83	24.75
	Jaiswal et al. (2025)*	23.17	26.31	50.67	21.33	0.00	26.87	24.73
	Zhou et al. (2025)*	41.30	52.61	72.14	40.61	11.22	29.85	41.29
	Dong et al. (2025)*	53.75	56.19	71.67	44.88	36.69	41.01	51.70
	Li et al. (2026b)	49.76	56.16	72.37	-	-	-	-
	Ours	**54.31**	56.51	**74.19**	**49.66**	**38.67**	**49.25**	**53.77**

Table 3: Performance and efficiency of our pruning approach under different sparsity levels (25% and 50%). We report the accuracy scores for Mixtral-8x7B and Mixtral-8x7B-Instruct, alongside the relative RAM usage and inference speedup compared to the unpruned baseline.

(a) Mixtral-8x7B

Sparsity	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.	RAM	Speedup
25%	59.85	62.14	75.62	52.56	44.20	60.80	59.20	$\times$0.76	$\times$1.20
50%	54.31	56.51	74.19	49.66	38.67	49.25	53.77	$\times$0.52	$\times$1.27

(b) Mixtral-8x7B-Instruct

Sparsity	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.	RAM	Speedup
25%	63.13	65.17	75.22	59.04	54.59	59.31	62.74	$\times$0.76	$\times$1.20
50%	57.66	60.42	73.01	53.18	30.10	46.74	53.52	$\times$0.52	$\times$1.27

ability to accurately identify and protect the model's core knowledge and routing logic. Second, our approach is not only effective on the base model (Mixtral-8x7B) but also demonstrates superior performance compared to the random baseline on the instruction-tuned model (Mixtral-8x7B-Instruct), where expert specialization is more pronounced. This proves that our approach is able to capture the critical contribution of experts to the model's overall representation learning, thus making it widely applicable to MoE model compression tasks across various training stages.

To further demonstrate the effectiveness of our approach, we compare the performance of our best model against existing MoE compression approaches for the Mixtral-8x7B model, with all approaches evaluated at 50% expert sparsity. The results on various benchmark datasets are reported in Table 2. Our approach consistently outperforms existing MoE compression approaches across nearly all datasets and achieves the highest average accuracy (Avg.). In particular, compared with our approach, Lee et al. (2025) prune experts by solely examining the similarity of expert weights and router outputs, while they fail to account for the varying importance of experts across different tasks; Lu et al. (2024) exhaustively enumerates expert combinations to calculate the reconstruction error, which results in prohibitively high computational complexity; Dong et al. (2025) utilizes router and output metrics for pruning but overlooks the reconstruction error of individual experts. And all these approaches employ uniform pruning across all layers, which ignores the inherent heterogeneity in the importance distribution of the expert networks layer-by-layer. To validate this heterogeneity, we visualize the layer-wise expert importance distribution in Appendix A.5. Our approach shows its superiority to the aforementioned studies since, while maintaining a computational overhead close to a single forward pass, we integrate three distinct types of information—perturbation (reconstruction error), routing decisions, and activation strength—specifically tailored for the target task. Furthermore, we achieve layer-wise non-uniform expert pruning through a path-finding strategy.

To further assess the generalizability of our approach across MoE architectures, we evaluate it on the more fine-grained Qwen3-30B-A3B model. Results are reported in Appendix A.4.

Table 4: Ablation study on the information integration modalities, conducted for the Mixtral-8x7B model. We analyze two weight metrics: "TI" (transition intensity) and "IS" (expert importance score). "✓" indicates the inclusion of the information modality, while "×" indicates its exclusion.

IS	TI	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
✓	✓	54.31	**56.51**	**74.19**	**49.66**	**38.67**	**49.25**	**53.77**
×	✓	54.01	53.90	72.85	46.07	38.06	46.98	51.98
✓	×	**58.77**	54.09	66.85	41.55	0.70	38.18	43.36

4.2 Effect of Expert Sparsity

To systematically assess how the pruning ratio (i.e., expert sparsity) influences model performance and efficiency, we extend our evaluation to multiple sparsity levels across different MoE architectures. Specifically, we evaluate our approach under 25% and 50% expert sparsity on Mixtral-8x7B and Mixtral-8x7B-Instruct. The complete results are reported in Table 3. The following are some observations. First, language understanding and knowledge-intensive tasks (e.g., MMLU, HellaSwag, MedQA) exhibit higher sensitivity to sparsity. Both Mixtral-8x7B and Mixtral-8x7B-Instruct show clear performance reduction along increased sparsity. An explanation is that these tasks depend on a broad set of experts, and higher pruning ratios may remove critical information pathways required for recalling distributed knowledge. Second, for certain tasks, increasing retained experts from 50% to 75% yields only marginal performance gains. For instance, in both Mixtral models, WinoGrande exhibit noticeably smaller performance improvements. This phenomenon suggests that these tasks rely on a relatively compact subset of experts, with core information predominantly captured by a small number of specialists. Consequently, even under a 50% sparsity level, the experts retained through our path planning already cover the essential computation routes, and the additional specialists preserved at 25% sparsity provide limited incremental benefit overall. Third, our approach demonstrates significant reductions in memory footprint and inference latency. We observe that at 50% sparsity, the model requires only 52% of the original RAM, effectively alleviating the deployment burden of large-scale MoE models. Simultaneously, we achieve inference speedups of $1.20\times$ and $1.27\times$ for 25% and 50% sparsity, respectively. These results highlight that our trajectory-driven pruning not only preserves essential capabilities but also serves as a viable solution for resource-constrained scenarios.

4.3 Ablation Study

To validate the effectiveness of our approach in information integration, we conduct an ablation study on the Mixtral-8x7B model to evaluate the impact of different information integration strategies used in the expert pruning approach. Specifically, we analyze the impact of two distinct weight metrics derived in Sec. 2.2: the transition intensity (TI) and the expert importance score (IS). To isolate the contribution of each weight, we evaluate variants where specific information is disregarded by setting the corresponding weight to a constant value of 1, thereby neutralizing its influence on the path planning and pruning process. The complete results are presented in Table 4. Several observations are made from these results. We observe that using only transition intensity provides a more balanced profile but fails to reach the peak average accuracy. This indicates that transition intensity focuses on preserving the topological structure of expert connections, but may fail to retain some experts that store essential knowledge within the model. Conversely, relying solely on the expert importance score leads to unstable performance; while it maintains high accuracy on a MMLU, it suffers a catastrophic failure in mathematical reasoning, significantly dragging down the average. This catastrophic failure suggests that merely capturing the static importance of individual experts is insufficient to identify and preserve the complex expert combination paths required for some specific tasks. Ultimately, the combination of both metrics achieves the highest average accuracy, confirming that TI and IS provide complementary information essential for maintaining model capabilities across diverse domains. Finally, we show results for the comparison between uniform and non-uniform pruning strategies in the Appendix A.1. To further investigate the role of the global path planning mechanism beyond the proposed metrics, we conduct an additional ablation comparing our full method against a path-free variant that greedily selects top experts based solely on local scores. As detailed in Appendix A.3,

the global planning is crucial for maintaining performance on reasoning-heavy tasks, confirming that preserving cross-layer trajectories is essential for complex reasoning.

5 Related Work

LLMs have achieved remarkable performance on many tasks in artificial intelligence (Brown et al., 2020; Li et al., 2020; Qin et al., 2022; Touvron et al., 2023; Gan et al., 2023; Liu et al., 2024b; Li et al., 2024; Yang et al., 2025). The MoE architecture has become a mainstream paradigm in LLMs, enabling exponential parameter scaling through sparse activation (Guo et al., 2025a; Cai et al., 2025). However, MoE models still require loading all expert weights during inference, resulting in a prohibitively substantial memory footprint that makes compression absolutely indispensable for practical deployment. Current MoE compression strategies fall into three major categories: dynamic expert selection, expert merging, and expert pruning. Dynamic expert selection introduces zero-computation experts or estimates token-level importance to dynamically adjust the number of activated experts during inference (Jin et al., 2024; Raposo et al., 2024; Zeng et al., 2024; Yue et al., 2024; Team et al., 2025b). However, since the total parameter count remains unchanged, these approaches do not fundamentally reduce memory usage, limiting their utility as true compression approaches. Expert merging reduces the number of experts by linearly combining weights, but the criteria it relies on—such as weight similarity—are often oversimplified and brittle (Li et al., 2023a; He et al., 2023; Chen et al., 2024; Li et al., 2025). As a result, merged experts frequently fail to preserve the specialized knowledge of their originals, causing unpredictable performance degradation. For expert pruning approaches, they differ from conventional pruning models (Frantar & Alistarh, 2023; Sun et al., 2023; Tian et al., 2025) since the expert pruning studies remove parameters at the expert-level, where all parameters in an expert are pruned if the expert is determined to be pruned (Yoon & Choi, 2026; Hu et al., 2026). These expert pruning studies initially rely on perturbation-based analysis, measuring output differences before and after removing each expert (Cao et al., 2024; Lu et al., 2024). Later studies propose more efficient metrics such as routing frequency and activation norms for expert selection (Muzio et al., 2024; Chowdhury et al., 2024; Dong et al., 2025; Lasby et al., 2025; Zhou et al., 2025; Liu et al., 2026). Recent studies further explore structure-aware or globally informed expert pruning through expert interaction graphs or Markov-chain routing dynamics (Li et al., 2026b; Goel et al., 2026). These developments have substantially improved pruning efficiency and performance, yet most approaches primarily evaluate expert importance using local or layer-wise statistics. Moreover, many prior pruning strategies adopt a uniform pruning ratio across layers, overlooking the heterogeneous contribution of experts in different layers. Different from these approaches, we directly formulate task-specific expert selection over complete trajectories in a layered weighted graph. Our approach integrates perturbation signals, routing decisions, and expert activations into a unified task-aware weighted graph for expert evaluation. By selecting experts through high-value trajectories, the framework adaptively determines the number of experts to retain in each layer, enabling efficient, fine-grained, and naturally non-uniform pruning.

6 Conclusion

In this paper, we presented a novel trajectory-driven expert pruning approach for domain-specific MoE model compression. By reformulating the MoE architecture as a directed weighted graph, our approach provides a global perspective on cross-layer expert dependencies. To achieve fine-grained expert evaluation, we introduced a multi-granularity weighting scheme that integrates perturbation-based loss, routing probabilities, and expert activations. Building on this representation, we employed a dynamic-programming-based global trajectory inference algorithm to identify critical computational trajectories and retain only the experts that lie along them. Experimental results demonstrate that our approach consistently outperforms existing pruning baselines across multiple benchmarks, while effectively preserving both general-domain and domain-specific capabilities. Overall, this work offers a unified, efficient, and interpretable solution for practical task-aware MoE compression.

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.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*, 2021.

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

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Mingyu Cao, Gen Li, Jie Ji, Jiaqi Zhang, Xiaolong Ma, Shiwei Liu, and Lu Yin. Condense, don't just prune: Enhancing efficiency and performance in moe layer pruning. *arXiv preprint arXiv:2412.00069*, 2024.

I Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, Chun-Yi Lee, et al. Retraining-free merging of sparse moe via hierarchical clustering. *arXiv preprint arXiv:2410.08589*, 2024.

Mohammed Nowaz Rabbani Chowdhury, Meng Wang, Kaoutar El Maghraoui, Naigang Wang, Pin-Yu Chen, and Christopher Carothers. A provably effective method for pruning experts in fine-tuned sparse mixture-of-experts. *arXiv preprint arXiv:2405.16646*, 2024.

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, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Zican Dong, Han Peng, Peiyu Liu, Wayne Xin Zhao, Dong Wu, Feng Xiao, and Zhifeng Wang. Domain-specific pruning of large mixture-of-experts models with few-shot demonstrations. *arXiv preprint arXiv:2504.06792*, 2025.

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*, pp. 10323–10337. PMLR, 2023.

Ruyi Gan, Ziwei Wu, Renliang Sun, Junyu Lu, Xiaojun Wu, Dixiang Zhang, Kunhao Pan, Junqing He, Yuanhe Tian, Ping Yang, et al. Ziya2: Data-centric learning is all llms need. *arXiv preprint arXiv:2311.03301*, 2023.

Palaash Goel, Ayush Maheshwari, and Tanmoy Chakraborty. It takes a maestro to prune bad experts. *arXiv preprint arXiv:2607.08601*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Hongcheng Guo, Juntao Yao, Boyang Wang, Junjia Du, Shaosheng Cao, Donglin Di, Shun Zhang, and Zhoujun Li. Cluster-driven expert pruning for mixture-of-experts large language models. *arXiv preprint arXiv:2504.07807*, 2025b.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. Merging experts into one: Improving computational efficiency of mixture of experts. *arXiv preprint arXiv:2310.09832*, 2023.

Shwai He, Daize Dong, Liang Ding, and Ang Li. Demystifying the compression of mixture-of-experts through a unified framework. *arXiv e-prints*, pp. arXiv–2406, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Wentao Hu, Mingkuan Zhao, Shuangyong Song, Xiaoyan Zhu, Xin Lai, and Jiayin Wang. Mosaic pruning: A hierarchical framework for generalizable pruning of mixture-of-experts models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21885–21893, 2026.

Ajay Jaiswal, Jianyu Wang, Yixiao Li, Pingzhi Li, Tianlong Chen, Zhangyang Wang, Chong Wang, Ruoming Pang, and Xianzhi Du. Finding fantastic experts in moes: A unified study for expert dropping strategies and observations. *arXiv preprint arXiv:2504.05586*, 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.

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

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moe++: Accelerating mixture-of-experts methods with zero-computation experts. *arXiv preprint arXiv:2410.07348*, 2024.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. *arXiv preprint arXiv:2510.13999*, 2025.

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

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Chenxi Li, Yuanhe Tian, Zhaxi Zerong, Yan Song, and Fei Xia. Challenging large language models with new tasks: A study on their adaptability and robustness. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 8140–8162, 2024.

Houyi Li, Ka Man Lo, Shijie Xuyang, Ziqi Wang, Wenzhen Zheng, Haocheng Zhang, Zhao Li, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Mixture-of-experts can surpass dense llms under strictly equal resource. In *International Conference on Learning Representations*, volume 2026, pp. 96868–96889, 2026a.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. *arXiv preprint arXiv:2004.14769*, 2020.

Lin Li, Yan Wang, and Zhuopeng Wang. C-gnn-prune: A unified graph-based framework for structure-aware pruning of mixture-of-experts models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22976–22984, 2026b.

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*, 2025.

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. *arXiv preprint arXiv:2310.01334*, 2023a.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023b.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Chang Liu, Yuanhe Tian, Weidong Chen, Yan Song, and Yongdong Zhang. Bootstrapping large language models for radiology report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18635–18643, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Zongfang Liu, Jinghui Zhang, Zijian Ma, Guangyi Chen, and Xin Yuan. How to score experts for one-shot moe expert pruning: A unified formulation and selection principle. *arXiv preprint arXiv:2606.15716*, 2026.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4427–4447, 2025.

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. *arXiv preprint arXiv:2402.14800*, 2024.

AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on*, 4(7):2025, 2025.

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

Han Qin, Yuanhe Tian, and Yan Song. Enhancing relation extraction via adversarial multi-task learning. In *Proceedings of the thirteenth language resources and evaluation conference*, pp. 6190–6199, 2022.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

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, 2021.

Zunhai Su, Qingyuan Li, Hao Zhang, YuLei Qian, Yuchen Xie, 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. *arXiv preprint arXiv:2306.11695*, 2023.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

Meituan LongCat Team, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, et al. Longcat-flash technical report. *arXiv preprint arXiv:2509.01322*, 2025b.

Meituan LongCat Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chen Gao, Chen Zhang, Chengcheng Han, et al. Longcat-flash-thinking-2601 technical report. *arXiv preprint arXiv:2601.16725*, 2026.

Yuanhe Tian, Junjie Liu, Xican Yang, Haishan Ye, and Yan Song. Frustratingly easy task-aware pruning for large language models. *arXiv preprint arXiv:2510.22489*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

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.

Seohee Yoon and Yong Suk Choi. Compel: Compensated mixture-of-experts pruning with expert-layer distribution. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30408–30423, 2026.

Tongtian Yue, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-k routing: Boosting the efficiency of moe-based llms. In *The Thirteenth International Conference on Learning Representations*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models. *arXiv preprint arXiv:2406.13233*, 2024.

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.

Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, Hehe Fan, et al. Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts llms. *arXiv preprint arXiv:2509.10377*, 2025.

A EXPERIMENT APPENDICES

Table 5: Comparison between uniform layer-wise expert pruning and our trajectory-driven non-uniform expert pruning strategy on Mixtral-8x7B at 50% expert sparsity. Both approches use the same metric, while differing in whether experts are pruned uniformly across layers or adaptively retained via global path planning.

Approach	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
Uniform	46.72	53.24	69.22	45.73	22.21	38.65	45.96
Non-uniform	**54.31**	**56.51**	**74.19**	**49.66**	**38.67**	**49.25**	**53.77**

Table 6: Performance comparison between the 1-expert and 2-expert path formulations on the Mixtral-8x7B model at 50% expert sparsity.

Approach	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
2-expert path	51.25	54.13	72.61	48.04	21.68	41.71	48.24
1-expert path	**54.31**	**56.51**	**74.19**	**49.66**	**38.67**	**49.25**	**53.77**

A.1 EFFECT OF NON-UNIFORM PRUNING

To further study the effect of layer-wise expert allocation, we compare uniform and non-uniform expert pruning strategies. For uniform pruning, experts are independently ranked within each MoE layer according to the $r_i^{(l)} \times e_i^{(l)} \times a_i^{(l)}$ score, and the experts with lower rankings are removed per layer. In contrast, our non-uniform pruning approach performs global trajectory-based expert selection, allowing different layers to retain different numbers of experts. Specifically, we evaluate both approaches under 50% expert retention on Mixtral-8x7B. The comparative results are reported in Table 5. Several critical observations can be drawn from the results. First, our approach consistently outperforms the uniform baseline across all datasets, achieving a 7.81% increase in average accuracy. This indicates that expert importance is distributed heterogeneously across layers; uniform pruning fails to allocate sufficient resources to critical layers, whereas our method adaptively retains experts where they are most needed. Second, the performance gap is most pronounced on reasoning-intensive tasks like GSM8K and MMLU. This suggests that complex reasoning relies on specific expert trajectories spanning multiple layers. Our trajectory-driven approach preserves these complete reasoning paths, which are otherwise disrupted by the rigid constraints of uniform layer-wise pruning.

A.2 EFFECT OF MULTI-EXPERT PATH FORMULATION

To investigate the alignment between our graph abstraction and the actual routing mechanism of models like Mixtral (which routes each token to two experts), we conduct an ablation study evaluating a "2-expert path" formulation. In this setup, we define the nodes in our computational graph as expert pairs. To construct these discrete nodes, we derive the true joint expert importance ($e_i^{(l)}$) by computing the L2 norm between the output of the expert pair and the original output. And we assign the routing probability ($r_i^{(l)}$) and activation ($a_i^{(l)}$) of the expert pair node by simply averaging the respective values of the individual experts within that pair. The complete results on the Mixtral-8x7B model at 50% expert sparsity are presented in Table 6. Empirically, we observe a counter-intuitive phenomenon. Contrary to the expectation of better matching the actual routing mechanism, the 2-expert path actually degrades performance across all evaluated benchmarks. Theoretically, this degradation occurs because simple averaging destroys the token-specific dynamic weights, inherently injecting combinatorial noise. Therefore, forcing a discrete joint-modeling of expert pairs is more combinatorial but not more faithful to the actual routing dynamics. Computationally, this multi-expert formulation triggers a severe combinatorial explosion. For Mixtral-8x7B, it expands the layer-wise state space from 8 to 28, surging the calibration overhead by 3.5×. Crucially, it completely fails on fine-grained MoEs like Qwen3-30B-A3B (routing 8 of 128 experts), requiring an intractable $\binom{128}{8} \approx 1.43 \times 10^{12}$ forward passes per layer. Consequently, our 1-expert path avoids both the theoretical noise and computational explosion, serving as a highly robust computational proxy to efficiently retain critical cross-layer information.

Table 7: Ablation study on the necessity of global path planning, conducted for the Mixtral-8x7B model at 50% expert sparsity. "✓" indicates the use of our global path planning, while "×" indicates a path-free baseline relying purely on global greedy selection.

Path Planning	MMLU	HellaSwag	WinoGrande	ARC	GSM8K	MedQA	Avg.
×	52.02	55.15	70.24	44.54	13.42	49.02	47.40
✓	**54.31**	**56.51**	**74.19**	**49.66**	**38.67**	**49.25**	**53.77**

Table 8: Performance comparison on the Qwen3-30B-A3B model at 50% expert sparsity. We evaluate across general knowledge (MMLU) and specific domains (GSM8K, MedQA).

Approach	MMLU	GSM8K	MedQA
Unpruned Original	77.79	88.86	72.82
Random Baseline	35.07	21.30	37.94
Ours	**46.61**	**56.03**	**45.01**

A.3 Necessity of Global Path Planning

To explicitly disentangle our proposed metric from the global path planning mechanism itself, we conduct a path-free ablation experiment. Specifically, this baseline utilizes our combined metric ($r_i^{(l)} \times e_i^{(l)} \times a_i^{(l)}$) to evaluate all experts, but removes the path search process, instead greedily retaining the top experts globally based purely on these local scores. The complete results are presented in Table 7. We can observe several critical phenomena from this comparison. First, the path-free baseline achieves respectable performance on general and knowledge-intensive tasks (e.g., MMLU, WinoGrande), which validates the fundamental quality of our proposed metric. This indicates that for tasks relying more on static knowledge retrieval, accurately capturing local statistics can provide a certain degree of effectiveness. However, our approach exhibits a massive advantage over the path-free baseline on the complex mathematical reasoning task (GSM8K). For multi-step mathematical reasoning, preserving an unbroken cross-layer information flow is vital. This severe degradation firmly proves that global path planning is indispensable for maintaining the model's deep reasoning capabilities.

A.4 Generalization to Qwen3-30B-A3B

To validate the generalization potential of our trajectory-driven framework on more recent and highly fine-grained MoE architectures, we extend our approach to the Qwen3-30B-A3B model. While keeping the total retained expert count consistent with our original settings, we naturally scale our multi-path routing top-k to 64 (with a non-overlapping constraint) to efficiently accommodate this massive expert space. We evaluate the pruned model at a 50% expert sparsity level across one general-domain benchmark (MMLU) and two specific-domain tasks (GSM8K and MedQA). As shown in Table 8, we observe that our method consistently and significantly outperforms the random pruning baseline across all evaluated tasks. This result firmly demonstrates that our approach is not limited to the Mixtral family, but possesses generalization capabilities for modern, highly fine-grained MoE architectures.

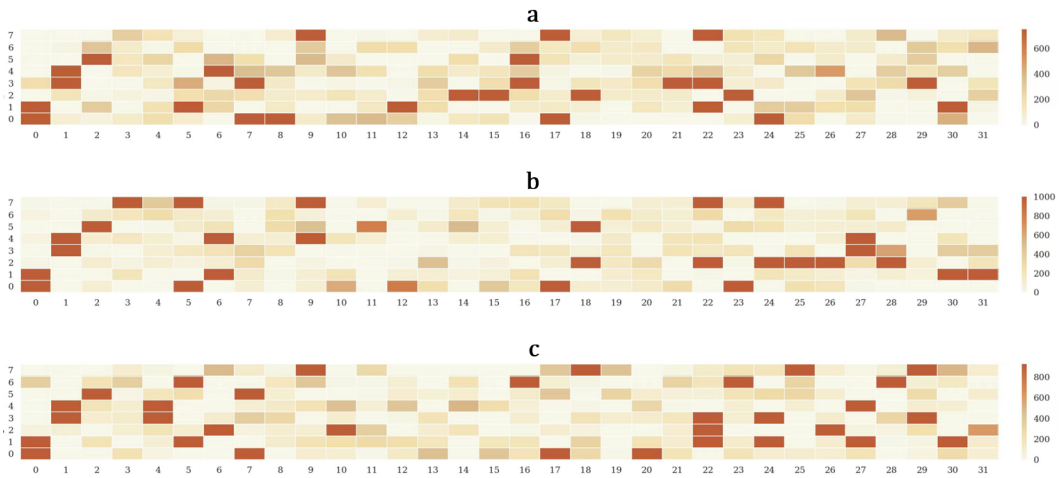

Figure 2: Visualization of expert importance for the Mixtral-8x7B model. These heatmaps illustrate the layer-wise and expert-specific importance, quantified by the frequency an expert is selected during the path planning phase. Calibration data are partitioned into $K = 10$ clusters, and expert frequencies are computed by sampling the top-100 paths for each cluster. An expert's importance is determined by its selection frequency, where a higher selection frequency (indicated by a darker color) signifies greater importance. The results include a general-domain task, MMLU (top), and two domain-specific tasks, MedQA (middle) and GSM8K (bottom).

A.5 Visualization of Expert Importance

To better understand expert specialization within MoE architectures, we quantify and visualize expert importance based on their selection frequency during the path planning phase. Specifically, we conduct experiments using the Mixtral-8x7B model. We perform clustering (K=10) on calibration data derived from three distinct benchmarks: MMLU, GSM8K, and MedQA. During the routing process, we record the selection frequency of each expert by sampling the top-m paths (where m=100). Experts identified as outliers in the data pre-processing stage were assigned the maximum selection count to reflect their highest potential importance. The aggregated results for each task are presented as heatmaps; in these visualizations, darker colors correlate with higher selection frequency, thereby indicating the relative importance of an expert for a given task. The results are presented in Figure 2. Two key findings are observed. First, the expert importance profiles are consistent across MMLU, MedQA, and GSM8K in the shallow layers (i.e., first to third layers). Conversely, a significant divergence is observed in the deep layers (i.e., 25th to 31st layers). Specific domain tasks (MedQA and GSM8K) demonstrate that critical importance is concentrated within a small subset of experts (e.g., MedQA's highest importance relies on the first and second experts in the deep blocks). In contrast, the general domain task MMLU maintains a more distributed importance pattern throughout the deeper layers. This indicates that the initial processing stages utilize a shared, domain-agnostic set of experts, and that the model achieves expert hyper-specialization only in the deeper layers. In these deep blocks, a few experts become disproportionately crucial for task-specific computation beyond foundational feature extraction. Second, at the 23rd layer, the first expert is crucial for the MedQA task (high selection frequency), but its contribution is negligible for GSM8K (selection frequency near zero). Similar phenomena are observed in several layers, such as the 18th, 25th, and 26th. This stark difference in the importance of the same expert across specific layers strongly suggests the orthogonality or task-isolation of expert functions within the MoE architecture.

A.6 Effect of the Number of Clusters

As described in section 3.1, our approach for constructing calibration data involves partitioning the dataset using k-means clustering and selecting representative samples. In this section, we investigate the impact of the number of clusters (K) on the pruned model's performance of the Mixtral-8x7B architecture. In particular, we prune the model to 50% expert sparsity using calibration data constructed with varying numbers of clusters, and visualize the resulting pruning performance.

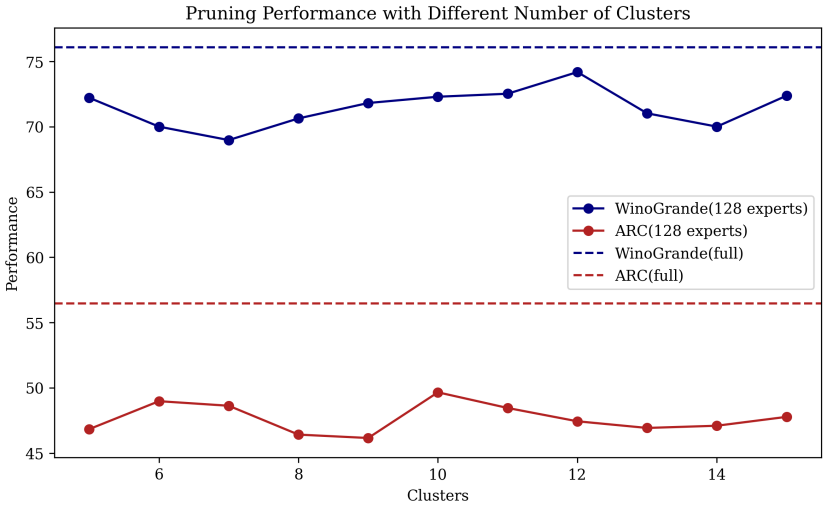

Figure 3: Effect of number of clusters (K) on the performance of the Mixtral-8x7B model pruned to 50% expert sparsity. The horizontal axis denotes the number of k-means clusters (K) used to construct calibration data, and the vertical axis shows the model's accuracy on WinoGrande (blue) and ARC (red).

The performance variations on WinoGrande and ARC are presented in Figure 3. The horizontal axis represents the number of k-means clusters (K) used to construct calibration data, and the vertical axis shows the pruned model's performance measured by accuracy. Several trends are observed. First, across both datasets, the pruned model exhibits a unimodal sensitivity to the choice of K, with performance increasing up to a critical point and deteriorating thereafter. This trend highlights the critical trade-off between the diversity and representativeness of the calibration data. When K is too small, the clustering is overly coarse, and the selected representative samples fail to capture the full diversity of the entire dataset, resulting in incomplete calibration gradient information. Conversely, when K is too large, the clustering becomes overly granular, potentially introducing noise or information redundancy, thereby diminishing the efficacy of the calibration information. Second, the optimal number of clusters differs between the two tasks (K=12 for WinoGrande and K=10 for ARC). This discrepancy reflects the inherent structural complexity and feature space heterogeneity of the different datasets, suggesting that building a high-quality calibration set is task-dependent.

B THEORETICAL APPENDICES

Table 9: The number of instances in the training, validation, and test sets of all datasets used in our experiments.

Dataset	Training	Validation	Test
MMLU	99,842	1,531	14,042
HellaSwag	39,905	10,042	10,003
WinoGrande	40,398	1,267	1,767
GSM8K	7,473	-	1,319
MedQA	10,178	-	1,273
ARC	1,119	299	1,172

B.1 DATASETS

To comprehensively evaluate our pruning approach, we conduct experiments on six widely used benchmark datasets that target different aspects of model performance. These datasets include MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC (Clark et al., 2018), GSM8K (Cobbe et al., 2021), and MedQA (Jin et al., 2021), covering both general and domain-specific fields.[5]

Specifically, MMLU is a multiple-choice benchmark covering 57 subjects (e.g., humanities and STEM), testing both factual knowledge and reasoning ability. HellaSwag is a commonsense reasoning benchmark where models select the most plausible sentence ending from multiple choices, known for being easy for humans but challenging for machines. WinoGrande is a large-scale dataset for assessing commonsense reasoning through pronoun resolution tasks, designed to minimize biases via debiasing techniques. GSM8K is a dataset of high-quality, linguistically diverse grade-school math word problems. MedQA is a medical question-answering dataset derived from United States Medical Licensing Examination questions in a four-choice format, covering clinical diagnosis and treatment scenarios in the medical domain. ARC contains grade-school science questions in multiple-choice format, requiring qualitative reasoning across physics, biology, and chemistry. The number of instances in the training, validation, and test datasets is reported in Table 9.

For the construction of the calibration set, we design and construct distinct calibration datasets tailored to tasks from various domains. Specifically, the calibration data for each task is derived from the training set of its corresponding dataset. To ensure sufficient diversity in the calibration data, we apply k-means clustering to each dataset independently, partitioning it into K clusters (the value of K is provided in Section 3.2). From each cluster, we select the sample closest to the cluster centroid to construct the representative calibration data. Furthermore, following the approach proposed by Dong et al. (2025), for each sample in the obtained calibration data, we replace its desired sample output by the output of an LLM with the sample input, which is demonstrated to be effective in LLM pruning.

[5]We obtain MMLU, HellaSwag, WinoGrande, ARC, GSM8K, and MedQA from `https://huggingface.co/datasets/cais/mmlu`, `https://huggingface.co/datasets/Rowan/hellaswag`, `https://huggingface.co/datasets/allenai/winogrande`, `https://huggingface.co/datasets/allenai/ai2_arc`, `https://huggingface.co/datasets/openai/gsm8k`, and `https://huggingface.co/datasets/GBaker/MedQA-USMLE-4-options`.

Algorithm 1 Trajectory-Based Expert Pruning

Require: Calibration dataset $\mathcal{D}$, Layers L, Experts N_e, Path budget m
Ensure: Experts to keep $\mathcal{E}_{\text{keep}}$

Initialize $\mathcal{E}_{\text{keep}} \leftarrow \emptyset$
for all sample $X \in \mathcal{D}$ **do**
  Compute $a_i^{(l)} \leftarrow \|\mathbf{H}_k^{(l-1)}(\mathbf{W}_i^{(l)})^\top\|_2$
  Compute $r_j^{(l+1)} \leftarrow [\mathrm{softmax}(\mathbf{H}_k^{(l)}(\mathbf{R}^{(l+1)})^\top)]_j$
  Compute $t_{i,j}^{(l)} \leftarrow a_i^{(l)} \times r_j^{(l+1)}$
  Compute $e_i^{(l)} \leftarrow \mathrm{softmax}(-\mathcal{L}_i^{(l)})$
  for $i = 1$ to N_e **do** ▷ Initialize path set
    $\mathcal{P}_i^{(1)} \leftarrow \{(\log e_i^{(1)}, [i])\}$
  end for
  for $l = 1$ to $L - 1$ **do** ▷ Iterative path update
    for $j = 1$ to N_e **do**
      $\mathcal{P}_{\text{next},j} \leftarrow \emptyset$
      for $i = 1$ to N_e **do**
        for all $(w_{\text{old}}, p) \in \mathcal{P}_i^{(l)}$ **do**
          $w_{\text{new}} \leftarrow w_{\text{old}} + \log t_{i,j}^{(l)} + \log e_j^{(l+1)}$
          $\mathcal{P}_{\text{next},j} \leftarrow \mathcal{P}_{\text{next},j} \cup \{(w_{\text{new}}, p \oplus j)\}$
        end for
      end for
      $\mathcal{P}_j^{(l+1)} \leftarrow \text{Top-}m(\mathcal{P}_{\text{next},j})$
    end for
  end for
  $\mathcal{P}_{m,X} \leftarrow \text{Top-}m\left(\bigcup_{i=1}^{N_e} \mathcal{P}_i^{(L)}\right)$
  $\mathcal{E}_{\text{keep},X} \leftarrow \bigcup_{p \in \mathcal{P}_{m,X}} \mathcal{V}(p)$
  $\mathcal{E}_{\text{keep}} \leftarrow \mathcal{E}_{\text{keep}} \cup \mathcal{E}_{\text{keep},X}$
end for

B.2 Detailed Pruning Algorithm

To provide a comprehensive view of our implementation, we present the complete trajectory-driven expert pruning procedure in Algorithm 1. This algorithm details the integration of activation strength, routing probabilities, and expert importance scores into the global path planning framework described in Section 2.

Table 10: Offline computational complexity of expert pruning approaches. All methods share the calibration forward-pass cost.

Approach	Data Collection	Metric Construction	Pruning Search
NAEE (Lu et al., 2024)	$O(C_{\text{fwd}})$	–	$O\Big(L \cdot \binom{N_e}{E} \cdot C_{\text{MoE}}(N, E)\Big)$
EASY-EP (Dong et al., 2025)	$O(C_{\text{fwd}})$	$O(N \cdot L \cdot K \cdot d)$	$O(L \cdot N_e \cdot \log N_e)$
Ours	$O(C_{\text{fwd}})$	$O(N \cdot L \cdot K \cdot d + S \cdot L \cdot N_e^2)$	$O(S \cdot L \cdot m \cdot N_e^2 \cdot \log(m+1))$

B.3 Computational Complexity Analysis

Our approach performs expert selection entirely offline, introducing no additional path-search overhead during inference. Let N and S denote the numbers of calibration tokens and samples, L the number of MoE layers, d the hidden dimension, N_e the number of experts per layer, K the number of activated experts per token, and m the number of prefixes retained per expert node. For NAEE, E denotes the number of retained experts per layer. We denote the forward-pass cost over the calibration dataset by C_{fwd}, and the cost of evaluating a candidate MoE layer with E experts over N tokens by $C_{\text{MoE}}(N, E)$. The resulting complexities are summarized in Table 10.

NAEE requires evaluating $\binom{N_e}{E}$ candidate expert subsets per layer, making its search combinatorially expensive. EASY-EP avoids this enumeration through activation-based statistics and layer-wise ranking. Our approach shares its $O(N \cdot L \cdot K \cdot d)$ token-level metric construction cost, while incorporating reconstruction loss, routing probabilities, and activation strengths from the same calibration forward pass. Transition matrix construction adds $O(S \cdot L \cdot N_e^2)$ operations, followed by a polynomial-time trajectory search with complexity $O(S \cdot L \cdot m \cdot N_e^2 \cdot \log(m+1))$. Although this search costs more than EASY-EP's layer-wise ranking, our approach achieves higher average accuracy (Table 2). Thus, it offers competitive pruning performance without combinatorial search or additional inference-time control.