EXPERTS EXHIBIT STRUCTURED REDUNDANCY: STRUCTURE-PRESERVING EXPERT PRUNING FOR SPARSE MIXTURE-OF-EXPERTS LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) large language models replace the dense feed-forward networks in standard Transformers with MoE layers consisting of multiple experts, activating only a small subset of experts for each token during the forward pass. This sparse activation mechanism significantly improves the trade-off between model capacity and inference efficiency. However, practical deployment still requires storing all experts, resulting in substantial memory overhead. In this work, we show that expert outputs in MoE layers exhibit structured redundancy. Across different models and task categories, the stable rank and entropy effective rank of expert outputs are consistently far below the nominal number of experts, while expert output distributions also exhibit fewer effective modes. Motivated by these observations, we reformulate expert pruning as a structure-preserving expert subset selection problem and efficiently optimize it via a greedy strategy, rather than only independently scoring and ranking individual experts or exhaustively searching over expert combinations. We propose two expert pruning methods under the unified structure-preserving framework: Projection Energy Expert Pruning (**PEEP**), which preserves the functional subspace by maximizing weighted projection energy, and Distributional Coverage Expert Pruning (**DCEP**), which preserves the distributional geometry by minimizing weighted coverage error. Both methods incorporate router-derived weights as priors, enabling them to jointly preserve the global structural relationships among experts while accounting for the local importance of individual experts. Experiments on DeepSeek-V2-Lite and Qwen3-30B-A3B demonstrate that, at the expert retention ratios of $r{=}75\%$ and $r{=}50\%$, our methods outperform the state-of-the-art baselines on the majority of benchmarks and consistently achieve the best average performance across both general tasks and mathematical reasoning tasks.

1 INTRODUCTION

Mixture-of-Experts (MoE) has become an important architectural paradigm for scaling large language models (LLMs) (Brown et al., 2020). Sparse MoE models replace the dense feed-forward networks (FFNs) in standard Transformers (Vaswani et al., 2017) with multiple experts and route each token to only a small subset of experts in each layer. This design enables scaling up the total number of model parameters and improving model capacity while keeping the number of activated parameters during inference. Despite their computational sparsity, MoE LLMs remain expensive to deploy, since all expert parameters must be stored during inference. As experts dominate the number of parameters, expert-level pruning becomes an increasingly important direction for MoE compression.

Existing expert pruning methods usually score and rank experts according to their individual importance based on activation frequency, router scores, activation magnitudes, or other hand-crafted metrics, and then remove the less important experts (Chen et al., 2022; Liu et al., 2024; Xie et al., 2024; Lasby et al., 2026). However, these approaches overlook that experts do not contribute to the model independently. The quality of a retained expert subset depends not only on the individual

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.

importance of each expert, but also on whether the selected experts jointly preserve the structural characteristics of the original MoE layer. Several recent studies have attempted to consider inter-expert interactions in expert pruning (Lu et al., 2024; Yang et al., 2024; Liu et al., 2024). However, they typically identify expert subsets through exhaustive enumeration or heuristic combinatorial optimization, introducing substantial pruning overhead. Such approaches are difficult to scale to modern MoE LLMs with tens or hundreds of experts per layer.

In this work, we reformulate expert pruning as a structure-preserving expert subset selection problem. Our empirical analysis in Figure 1 reveals that expert outputs in MoE layers exhibit strong structured redundancy. From a functional subspace perspective, both the stable rank and entropy effective rank of expert output representations are substantially lower than the number of experts. From a distributional perspective, expert output distributions also contain fewer effective modes than the number of experts. These observations motivate a new pruning framework that selects a compact yet complementary expert subset to preserve the structural characteristics of the original MoE layer.

We instantiate this framework with two expert pruning methods that avoid exhaustive enumeration and costly heuristic search. Projection Energy Expert Pruning (PEEP) views expert outputs as basis functions and greedily selects experts that maximize weighted projection energy, thereby preserving the dominant functional subspace structure. Distributional Coverage Expert Pruning (DCEP) represents expert outputs as Gaussian distributions and greedily selects medoid experts that minimize weighted coverage error, thereby preserving distributional geometry. Both methods incorporate router-derived weights as priors, enabling them to jointly preserve global structural relationships among experts and local importance of individual experts.

Our contributions are summarized as follows:

- We reveal structured redundancy in expert outputs from both functional subspace and distributional geometry perspectives, showing that the effective structure of expert representations is substantially smaller than the total number of experts.
- Unlike prior pruning methods that rank experts individually, we reformulate expert pruning as a structure-preserving expert subset selection problem and efficiently solve it without resorting to expensive combinatorial search over expert subsets.
- We instantiate the structure-preserving framework with two expert pruning methods, PEEP and DCEP. The former preserves the functional subspace structure, while the latter preserves the distributional geometry. Both methods are implemented through efficient greedy selection.
- Experiments on DeepSeek-V2-Lite and Qwen3-30B-A3B demonstrate that our methods outperform the state-of-the-art (SOTA) baselines on the majority of benchmarks and consistently achieve the best average performance across both general tasks and mathematical reasoning tasks.

2 Related Work

MoE Architectures MoE architectures scale model capacity while routing each token to a small subset of experts. Early work introduced learned sparse routing for conditional computation (Shazeer et al., 2017), followed by GShard (Lepikhin et al., 2021) and Switch Transformer (Fedus et al., 2022), which improved large-scale training. Mixtral (Jiang et al., 2024) showed that sparse MoE can deliver competitive performance with markedly lower inference cost. Recent MoE LLMs further enhance expert specialization through shared experts in DeepSeekMoE (Dai et al., 2024) and fine-grained experts in Qwen3 (Yang et al., 2025).

MoE Compression MoE compression reduces the number of parameter and memory overhead introduced by storing all experts. Expert pruning directly removes complete experts. Many existing methods score and rank experts according to their individual importance based on routing or activation statistics. TSEP performs task-specific progressive pruning during fine-tuning (Chen et al., 2022). Subsequently, SEER-MoE combines routing statistics with regularized fine-tuning (Muzio et al., 2024), while MC-Suite systematically investigates a broad range of expert importance metrics (Jaiswal et al., 2025). EASY-EP estimates domain-specific expert importance using router scores and similarity (Dong et al., 2025) of expert output, whereas REAP combines router scores

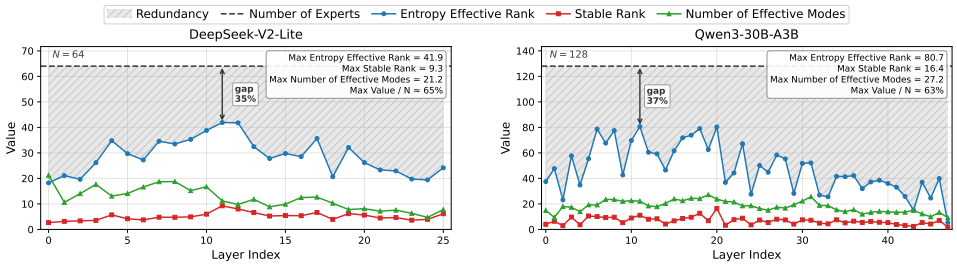

Figure 1: Ranks of expert representations and numbers of effective modes of expert output distributions across MoE layers. *Left:* DeepSeek-V2-Lite with 64 experts per layer. *Right:* Qwen3-30B-A3B with 128 experts per layer. Across all MoE layers, the entropy effective rank, stable rank, and number of effective modes remain well below the nominal number of experts, reaching at most 63%–65% of the total number of experts and thereby revealing substantial structured redundancy in expert outputs.

with activation norms (Lasby et al., 2026). Although efficient, these methods primarily rank experts independently. Other expert pruning methods consider inter-expert interactions more explicitly. NAEE selects expert combinations by minimizing layer-output reconstruction error through subset enumeration (Lu et al., 2024). EEP and MoE-I^2 employ evolutionary or genetic search to optimize the retained expert subsets (Liu et al., 2024; Yang et al., 2024). Shapley-MoE (Huang et al., 2026) estimates each expert's contribution using Shapley values and removes experts with relatively small contributions. These methods capture inter-expert interactions but may incur substantial search costs. Beyond pruning, STUN clusters experts according to behavioral similarity and then applies expert merging and unstructured pruning (Lee et al., 2025). MC-SMoE aligns and merges experts using routing-derived clusters (Li et al., 2024), HC-SMoE performs expert merging through hierarchical clustering (Chen et al., 2025), ConDense-MoE condenses sparse MoE layers into compact dense layers (Cao et al., 2026), and MoNE replaces redundant experts with lightweight novice modules (Zhang et al., 2025). Unlike methods that score and rank experts individually, search over possible expert subsets, or merge experts by modifying their parameters, our methods efficiently select and retain a subset of the original experts.

3 PRELIMINARIES

MoE Layer A standard Transformer layer contains a self-attention module followed by a FFN. In a sparse MoE Transformer, the dense FFN is replaced by N expert FFNs, denoted as E_i. Given the hidden state $\mathbf{h}_x \in \mathbb{R}^d$ of a token x, a linear router $\mathbf{W}_{\text{router}}$ computes the logit for each expert:

$$s_i = [\mathbf{W}_{\text{router}}\mathbf{h}_x]_i, \mathbf{W}_{\text{router}} \in \mathbb{R}^{N\times d}. \tag{1}$$

To reduce computation, the MoE layer activates only the experts with the highest top-k logits and normalizes them into routing scores

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j\in\mathcal{T}_x}\exp(s_j)}, \tag{2}$$

where $\mathcal{T}_x$ is the selected expert indices. The output of the MoE layer is then computed as a weighted sum of the outputs of the selected experts

$$\mathbf{h}_x^{'} = \sum_{i\in\mathcal{T}_x}\alpha_i E_i(x), \tag{3}$$

Expert Pruning Expert pruning aims to permanently remove a subset of experts from each MoE layer. Formally, for layer ℓ with N experts and a target retention ratio r, the goal is to select a retained expert subset

$$S^{(\ell)} \subseteq \{1,\dots,N\}, \quad |S^{(\ell)}| = K = \lfloor rN \rfloor, \tag{4}$$

and remove the other experts.

4 MOTIVATION

The output of an MoE layer is produced by a routed aggregation of multiple expert outputs, suggesting that expert redundancy can be understood in terms of the global structure formed by the outputs of all experts. We therefore investigate whether the expert outputs exhibit structural redundancy from two perspectives: *functional subspace* and *distributional geometry*.

Calibration Statistics Given a calibration dataset $\mathcal{D}$, we perform a forward pass through the model and collect the output of all experts. For the i-th expert in MoE layer ℓ, let $f_i^{(\ell)}(\mathbf{h}_x)$ denote the output for the hidden state $\mathbf{h}_x$ of token x. We estimate the mean of each expert's output as

$$\boldsymbol{\mu}_i^{(\ell)} = \mathbb{E}_{x\sim\mathcal{D}}\left[f_i^{(\ell)}(\mathbf{h}_x)\right]. \tag{5}$$

We additionally estimate the standard deviation of each expert's output as

$$\boldsymbol{\sigma}_i^{(\ell)} = \sqrt{\mathbb{E}\left[\mathbf{z}_i^2\right] - \left(\mathbb{E}\left[\mathbf{z}_i\right]\right)^2}, \mathbf{z}_i = f_i^{(\ell)}(\mathbf{h}_x). \tag{6}$$

Low-Dimensional Functional Subspace We use the mean of each expert's output as its representation and investigate whether the expert output representations are concentrated in a lower-dimensional subspace. For each MoE layer ℓ with N experts, we construct the representation matrix and compute its singular value decomposition

$$\mathbf{W}^{(\ell)} = \left[(\boldsymbol{\mu}_1^{(\ell)})^\top \quad \cdots \quad (\boldsymbol{\mu}_N^{(\ell)})^\top\right]^\top = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top, \mathbf{W}^{(\ell)} \in \mathbb{R}^{N\times d}. \tag{7}$$

We get the singular values $\sigma_1 \geq \sigma_2 \geq \cdots$. Then we quantify the effective dimensionality of expert output representations using the entropy effective rank based on normalized spectral energy

$$r_{\text{entropy}} = \exp\left(-\sum_j p_j \log p_j\right), p_j = \frac{\sigma_j^2}{\sum_k \sigma_k^2}, \tag{8}$$

and the stable rank

$$r_{\text{stable}} = \frac{\|\mathbf{W}^{(\ell)}\|_F^2}{\|\mathbf{W}^{(\ell)}\|_2^2} = \frac{\sum_j \sigma_j^2}{\sigma_1^2}. \tag{9}$$

Both metrics characterize the effective dimensionality of the subspace spanned by the expert output representations and are less sensitive than the algebraic rank to small singular values arising from numerical noise.

As shown in Figure 1, both metrics are substantially lower than the number of experts across all MoE layers. These results indicate that the expert output representations are in a lower-dimensional subspace, suggesting substantial redundancy in the function space spanned by the expert outputs. Therefore, it may be unnecessary to preserve all experts to maintain the function space. Instead, a compact subset of complementary experts may suffice to preserve the original function space.

Redundant Distributional Geometry The mean of each expert's output does not capture the distributional variation. We therefore examine the distributional geometry formed by the expert outputs and investigate whether different experts have similar distributional modes. We approximate the output distribution of expert i by a diagonal Gaussian

$$P_i^{(\ell)} = \mathcal{N}\left(\boldsymbol{\mu}_i^{(\ell)}, \text{diag}\left((\boldsymbol{\sigma}_i^{(\ell)})^2\right)\right). \tag{10}$$

For two experts i and j, we measure their distributional discrepancy using the squared 2-Wasserstein distance

$$d_{i,j}^{(\ell)} = \|\boldsymbol{\mu}_i^{(\ell)} - \boldsymbol{\mu}_j^{(\ell)}\|_2^2 + \|\boldsymbol{\sigma}_i^{(\ell)} - \boldsymbol{\sigma}_j^{(\ell)}\|_2^2. \tag{11}$$

We convert pairwise distances into a similarity kernel

$$K_{i,j}^{(\ell)} = \exp\left(-\frac{d_{i,j}^{(\ell)}}{c\cdot\tau^{(\ell)}}\right), \tag{12}$$

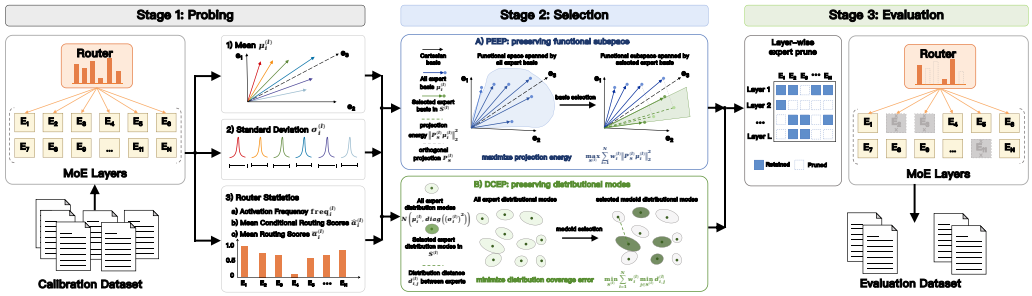

Figure 2: Overview of structure-preserving MoE expert pruning. *Probing:* collect expert outputs and router statistics on calibration dataset. *Selection:* PEEP maximizes projection energy; DCEP minimizes distributional coverage error. *Evaluation:* prune experts, restrict routing to the retained expert set and evaluate the pruned MoE LLMs on downstream tasks.

where $\tau^{(\ell)}$ is the median of the pairwise distances within layer and c is the scaling factor. The numbers of effective modes under different values of c are provided in Figure 7 in the appendix.

To estimate the effective number of distinct distributional modes, let $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of $\mathbf{K}^{(\ell)}$. We compute their participation ratio as

$$N_{\text{eff}}^{(\ell)} = \frac{\left(\sum_{j=1}^{N} \lambda_j\right)^2}{\sum_{j=1}^{N} \lambda_j^2}. \tag{13}$$

If expert distributions are mutually distinct, the similarity kernel approaches the identity matrix and $N_{\text{eff}^{(\ell)}}$ approaches N. In contrast, strongly similar expert distributions induce a spectrally concentrated kernel and yield $N_{\text{eff}}^{(\ell)} \ll N$. As shown in Figure 1, the effective number of distributional modes is consistently much smaller than the total number of experts. These results indicate that expert distributions contain a limited number of representative modes, with multiple experts providing overlapping coverage. A compact subset of experts is sufficient to preserve most of the coverage of the original expert distribution.

5 Method

The observations in the previous section reveal two forms of structured redundancy in MoE layers. First, the expert output representations are concentrated in a low-dimensional functional subspace, indicating that a subset of complementary experts can preserve the original function space. Second, expert output distributions exhibit a limited number of effective modes, suggesting that a subset of complementary experts is sufficient to capture the original distributional geometry. These findings motivate us to reformulate expert pruning as a *structure-preserving subset selection* problem, where the goal is to identify a compact expert subset that maximally preserves the structure of the original MoE layer outputs. We instantiate this structure-preserving framework with PEEP and DCEP. Both methods select expert subsets that minimize structural deviation from the original expert set. Both methods use a greedy strategy, thereby enabling effective expert pruning. An overview of our methods is shown in Figure 2.

5.1 Projection Energy Expert Pruning

Optimization Objective For layer ℓ, let $w_i^{(\ell)} \in \mathbb{R}^+$ denote router-derived weights for expert i in layer ℓ, we can define the weighted representation Gram matrix

$$\mathbf{C}^{(\ell)} = \left(\mathbf{W}^{(\ell)}\right)^\top \text{diag}\left(\boldsymbol{w}^{(\ell)}\right) \left(\mathbf{W}^{(\ell)}\right), \tag{14}$$

where $\mathbf{W}^{(\ell)}$ is the expert representation matrix in Equation equation 7, $\boldsymbol{w}^{(\ell)} = [w_1^{(\ell)}, ..., w_N^{\ell}]^\top \in \mathbb{R}_+^N$. For a retained expert set $S^{(\ell)} \subseteq \{1, \ldots, N\}$, let $\mathbf{P}_S^{(\ell)}$ be the orthogonal projector onto $\mathcal{W}_S^{(\ell)} =$

Algorithm 1 Projection Energy Expert Pruning

Input: expert representation matrix $\mathbf{W}^{(\ell)} \in \mathbb{R}^{N\times d}$, weights $\boldsymbol{w}^{(\ell)} \in \mathbb{R}^N$, number of retained experts K, tolerance ϵ
Output: retained expert subset $S^{(\ell)}$

Normalize $\boldsymbol{w}^{(\ell)} \leftarrow \boldsymbol{w}^{(\ell)}/\mathrm{mean}(\boldsymbol{w}^{(\ell)})$
$\mathbf{C}^{(\ell)} \leftarrow \big(\mathbf{W}^{(\ell)}\big)^\top \mathrm{diag}(\boldsymbol{w}^{(\ell)})\,\mathbf{W}^{(\ell)}$
$S^{(\ell)} \leftarrow \emptyset;\ \mathbf{Q} \leftarrow [\,] \in \mathbb{R}^{d\times 0}$
for $t = 1$ **to** K **do**
  $\Delta^\star \leftarrow -\infty;\ j^\star \leftarrow \texttt{null};\ \boldsymbol{q}^\star \leftarrow \texttt{null}$
  for each $j \in \{1,\dots,N\} \setminus S^{(\ell)}$ **do**
    $\boldsymbol{v} \leftarrow \boldsymbol{\mu}_j^{(\ell)}$
    $\boldsymbol{v}_\perp \leftarrow \boldsymbol{v} - \mathbf{Q}\mathbf{Q}^\top \boldsymbol{v}$
    if $\|\boldsymbol{v}_\perp\|_2 > \epsilon$ **then**
      $\boldsymbol{q} \leftarrow \boldsymbol{v}_\perp/\|\boldsymbol{v}_\perp\|_2$
      $\Delta_j \leftarrow \boldsymbol{q}^\top \mathbf{C}^{(\ell)} \boldsymbol{q}$
      if $\Delta_j > \Delta^\star$ **then**
        $\Delta^\star \leftarrow \Delta_j;\ j^\star \leftarrow j;\ \boldsymbol{q}^\star \leftarrow \boldsymbol{q}$
      end if
    end if
  end for
  $S^{(\ell)} \leftarrow S^{(\ell)} \cup \{j^\star\};\ \mathbf{Q} \leftarrow [\mathbf{Q}, \mathbf{q}^\star]$
end for
return $S^{(\ell)}$

$\mathrm{span}\{\boldsymbol{\mu}_i^{(\ell)} : i \in S^{(\ell)}\}$. PEEP selects experts by minimizing the weighted projection error

$$\mathcal{L}_{\mathrm{PEEP}}(S^{(\ell)}) = \sum_{i=1}^{N} w_i^{(\ell)} \big\|\boldsymbol{\mu}_i^{(\ell)} - \mathbf{P}_S^{(\ell)} \boldsymbol{\mu}_i^{(\ell)}\big\|_2^2, \tag{15}$$

which is equivalent to maximizing the weighted projection energy $\sum_{i=1}^N w_i^{(\ell)} \big\|\mathbf{P}_S^{(\ell)} \boldsymbol{\mu}_i^{(\ell)}\big\|_2^2$. Intuitively, PEEP seeks a subset of complementary basis experts whose span approximates the original function space.

Greedy Selection We adopt a greedy procedure that iteratively adds the expert yielding the largest marginal energy gain. Let $\mathbf{Q} \in \mathbb{R}^{d\times t}$ be an orthonormal basis of $\mathcal{W}_S^{(\ell)}$ after t selections. For candidate $j \notin S^{(\ell)}$, define the residual direction

$$\boldsymbol{q}_j \ \propto\ \big(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top\big)\boldsymbol{\mu}_j^{(\ell)}, \qquad \|\boldsymbol{q}_j\|_2 = 1, \tag{16}$$

which removes components of $\boldsymbol{\mu}_j^{(\ell)}$ already explained by $S^{(\ell)}$. The marginal gain of adding expert j is

$$\Delta_{\mathrm{PEEP}}(j \mid S^{(\ell)}) = \sum_{i=1}^{N} w_i^{(\ell)} \big(\boldsymbol{q}_j^\top \boldsymbol{\mu}_i^{(\ell)}\big)^2 = \boldsymbol{q}_j^\top \mathbf{C}^{(\ell)} \boldsymbol{q}_j. \tag{17}$$

At each step we choose $j^\star = \arg\max_{j\notin S^{(\ell)}} \Delta_{\mathrm{PEEP}}(j \mid S^{(\ell)})$, update $\mathbf{Q} \leftarrow [\mathbf{Q}, \mathbf{q}_{j^\star}]$, and repeat until $|S^{(\ell)}| = K$, where K is the number of retained experts in Equation equation 4. Algorithm 1 summarizes the PEEP procedure.

5.2 DCEP: Distributional Coverage Expert Pruning

Optimization Objective For layer ℓ, let $w_i^{(\ell)} \in \mathbb{R}_+$ denote router-derived weights for expert i in layer ℓ. According to the distance matrix in Equation equation 11, we define the coverage error of each expert i by the distance to its nearest medoid in selected expert set $S^{(\ell)}$. DCEP selects experts by minimizing the weighted coverage error

$$\mathcal{L}_{\mathrm{DCEP}}(S^{(\ell)}) = \sum_{i=1}^{N} w_i^{(\ell)} \min_{j\in S^{(\ell)}} d_{i,j}^{(\ell)}. \tag{18}$$

Algorithm 2 Distributional Coverage Expert Pruning

Input: mean of each expert's output $\{\boldsymbol{\mu}_i^{(\ell)}\}$, std of each expert's output $\{\boldsymbol{\sigma}_i^{(\ell)}\}$, weights $\boldsymbol{w}^{(\ell)}$, number of retained experts K

Output: retained expert subset $S^{(\ell)}$

Z-score normalize $\boldsymbol{\mu}_i^{(\ell)}, \boldsymbol{\sigma}_i^{(\ell)}$ across experts for each dimension
Calculate $d_{i,j}^{(\ell)}$ via Eq. equation 11
Normalize $\boldsymbol{w}^{(\ell)} \leftarrow \boldsymbol{w}^{(\ell)}/\mathrm{mean}(\boldsymbol{w}^{(\ell)})$
$S^{(\ell)} \leftarrow \emptyset$; $\delta_i \leftarrow +\infty$ for $i = 1, \ldots, N$
for $t = 1$ **to** K **do**
  $\Delta^\star \leftarrow -\infty$; $j^\star \leftarrow$ `null`
  for each $j \in \{1, \ldots, N\} \setminus S^{(\ell)}$ **do**
    if $t = 1$ **then**
      $\Delta_j \leftarrow -\sum_{i=1}^{N} w_i^{(\ell)} d_{i,j}^{(\ell)}$
    else
      $\Delta_j \leftarrow \sum_{i=1}^{N} w_i^{(\ell)} \left[\delta_i - \min(\delta_i, d_{i,j}^{(\ell)})\right]$
    end if
    if $\Delta_j > \Delta^\star$ **then**
      $\Delta^\star \leftarrow \Delta_j$; $j^\star \leftarrow j$
    end if
  end for
  $S^{(\ell)} \leftarrow S^{(\ell)} \cup \{j^\star\}$; $\delta_i \leftarrow \min(\delta_i, d_{i,j^\star}^{(\ell)})$ for $i = 1, \ldots, N$
end for
return $S^{(\ell)}$

Intuitively, DCEP seeks a subset of medoid experts whose distributional modes approximately cover the original distributional geometry.

Greedy Selection We also adopt a greedy procedure that iteratively adds the expert yielding the largest coverage gain. Let $\delta_i(S^{(\ell)}) = \min_{j \in S^{(\ell)}} d_{i,j}^{(\ell)}$ be the current nearest-medoid distance for expert i, with $\delta_i(\emptyset) = +\infty$. When adding candidate $j \notin S^{(\ell)}$, the coverage gain is

$$\Delta_{\mathrm{DCEP}}(j \mid S^{(\ell)}) = \sum_{i=1}^{N} w_i^{(\ell)} \Big[\delta_i(S^{(\ell)}) - \min\big(\delta_i(S^{(\ell)}),\, d_{i,j}^{(\ell)}\big)\Big]. \tag{19}$$

We select $j^\star = \arg\max_{j \notin S^{(\ell)}} \Delta_{\mathrm{DCEP}}(j \mid S^{(\ell)})$, update $\delta_i(S^{(\ell)}) \leftarrow \min(\delta_i(S^{(\ell)}), d_{i,j^\star}^{(\ell)})$, and repeat until $|S^{(\ell)}| = K$. For $S^{(\ell)} = \emptyset$, $\Delta_{\mathrm{DCEP}}(j \mid \emptyset) = -\sum_i w_i^{(\ell)} d_{i,j}^{(\ell)}$. Algorithm 2 summarizes the DCEP procedure.

5.3 Router-Derived Weights

Both PEEP and DCEP use the weights $w_i^{(\ell)}$ as a prior information of expert importance. We compute three router signals in the calibration forward pass: **(i)** *activation frequency* $freq_i^{(\ell)}$, the fraction of tokens for which expert i appears in $\mathcal{T}(x)$; **(ii)** *mean conditional routing score* $\hat{g}_i^{(\ell)}$, the average router score of expert i over the tokens for which expert i appears in $\mathcal{T}(x)$; **(iii)** *mean routing score* $\bar{g}_i^{(\ell)}$, the average router score of expert i over all tokens. Each signal is normalized to $(0, 1]$ within the layer and multiplicatively combined, e.g., $w_i^{(\ell)} \propto freq_i^{(\ell)} \hat{g}_i^{(\ell)} \bar{g}_i^{(\ell)}$ for the full combination. Weights are also normalized to unit mean. Router-derived weights reflect the local importance of individual experts, complementing PEEP and DCEP which capture the global structure of expert outputs.

6 Experiments

6.1 Experimental Setup

We evaluate our methods on DeepSeek-V2-Lite and Qwen3-30B-A3B at expert retention ratio of r=75% and r=50%. We assess general capability on HellaSwag (Zellers et al., 2019), ARC-Challenge (Clark et al., 2018), MMLU (Hendrycks et al., 2021a), and C-Eval (Huang et al., 2023),

and mathematical reasoning capability on GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), AIME24 (Mathematical Association of America, 2024), and AIME25 (Mathematical Association of America, 2025). Calibration uses 32 C4 samples (Raffel et al., 2020) for general tasks, 32 GSM8K training samples for GSM8K and MATH500, and the 30 AIME23 problems (Mathematical Association of America, 2023) for AIME24 and AIME25. The baselines include random pruning, frequency-based pruning, EASY-EP (Dong et al., 2025) and REAP (Lasby et al., 2026). Detailed experimental information is provided in Appendix A.

Table 1: Performance comparison of PEEP and DCEP with baseline expert pruning methods on DeepSeek-V2-Lite and Qwen3-30B-A3B across general and mathematical reasoning benchmarks at expert retention ratios of r=75% and r=50%. The best and second-best results among pruned methods under each model and expert retention ratio are highlighted in bold and underlined, respectively. "–" indicates that the model fails to complete the corresponding task.

r	Method	General					Mathematical Reasoning				
		HellaSwag	ARC-Challenge	MMLU	C-Eval	Average	GSM8K	MATH500	AIME24	AIME25	Average
DeepSeek-V2-Lite											
100%	Unpruned	77.7	46.0	55.5	57.8	59.3	40.8	16.2	–	–	28.5
75%	Random	60.1	35.5	36.3	35.7	41.9	9.9	4.8	–	–	7.4
	Frequency	74.3	40.5	36.2	29.6	45.2	36.5	**13.2**	–	–	24.9
	EASY-EP	65.5	37.5	35.5	35.1	43.4	25.7	6.4	–	–	16.1
	REAP	74.5	42.4	**45.7**	41.5	51.0	34.7	9.0	–	–	21.9
	PEEP (Ours)	**76.1**	**43.3**	45.2	39.5	51.0	39.9	**13.2**	–	–	26.5
	DCEP (Ours)	**76.1**	41.8	44.3	**42.7**	**51.2**	**40.8**	12.6	–	–	**26.7**
50%	Random	37.8	24.1	24.8	23.6	27.6	2.2	1.3	–	–	1.8
	Frequency	46.7	25.3	24.0	22.8	29.7	20.6	6.0	–	–	13.3
	EASY-EP	58.4	34.6	25.0	25.0	35.8	6.6	2.8	–	–	4.7
	REAP	**67.3**	34.1	25.0	24.2	37.7	11.2	3.0	–	–	7.1
	PEEP (Ours)	61.7	**36.8**	**28.1**	**26.7**	**38.3**	**23.4**	**6.2**	–	–	**14.8**
	DCEP (Ours)	62.4	34.1	26.8	25.6	37.2	21.8	5.8	–	–	13.8
Qwen3-30B-A3B											
100%	Unpruned	79.6	63.0	80.2	86.2	77.3	86.4	90.0	73.3	70.0	79.9
75%	Random	72.2	57.0	71.6	**73.4**	68.6	64.5	79.8	26.7	24.7	48.9
	Frequency	79.6	**62.5**	74.5	68.3	71.2	88.3	83.0	63.3	60.0	73.7
	EASY-EP	77.2	61.3	61.3	61.0	65.2	71.2	86.6	60.0	60.0	69.5
	REAP	79.4	60.7	75.1	70.0	71.3	86.9	**87.0**	66.7	60.0	75.1
	PEEP (Ours)	**80.5**	60.6	76.0	68.3	71.4	**88.9**	85.6	**76.7**	**63.3**	**78.6**
	DCEP (Ours)	80.2	62.1	**76.3**	68.2	**71.7**	**88.9**	86.6	70.0	60.0	76.4
50%	Random	56.2	38.9	51.5	**46.3**	48.2	30.1	13.6	0.0	0.0	10.9
	Frequency	42.7	27.1	24.6	23.0	29.4	87.8	27.4	53.3	43.3	53.0
	EASY-EP	70.9	**52.0**	49.3	35.7	52.0	39.1	19.2	**60.0**	50.0	42.1
	REAP	72.6	47.7	58.8	44.6	55.9	84.4	48.0	50.0	46.7	57.3
	PEEP (Ours)	**76.1**	46.6	59.7	44.4	**56.7**	**89.7**	40.6	50.0	50.0	57.6
	DCEP (Ours)	75.9	45.5	**60.0**	44.9	56.6	88.9	**51.6**	50.0	**53.3**	**60.9**

Table 2: Ablation study of router-derived weights on DeepSeek-V2-Lite and Qwen3-30B-A3B. Columns $freq_i^{(\ell)}$, $\hat{g}_i^{(\ell)}$, and $\bar{g}_i^{(\ell)}$ indicate whether activation frequency, mean conditional routing score, and mean routing score are included in the multiplicative weight construction. A ✓ indicates that the corresponding router signal is included, whereas a × indicates that it is excluded. For each model, performance is averaged over all evaluated tasks. The best and second-best results in each column are highlighted in bold and underlined, respectively.

Router Signals			DeepSeek-V2-Lite				Qwen3-30B-A3B			
$freq_i^{(\ell)}$	$\hat{g}_i^{(\ell)}$	$\bar{g}_i^{(\ell)}$	75%		50%		75%		50%	
			PEEP	DCEP	PEEP	DCEP	PEEP	DCEP	PEEP	DCEP
×	×	×	**43.6**	42.6	29.8	29.4	70.0	70.0	46.1	51.0
✓	✓	✓	42.2	**43.9**	30.0	30.4	**75.5**	**75.0**	57.7	58.3
✓	×	×	42.9	43.1	30.5	29.4	75.0	74.0	57.1	**58.8**
×	✓	×	43.2	43.0	24.9	29.4	70.2	72.4	55.4	54.8
×	×	✓	42.9	42.4	**31.1**	**30.7**	73.7	73.8	57.0	**58.8**
✓	✓	×	42.0	42.0	30.6	30.1	71.3	71.0	57.8	58.0
✓	×	✓	43.2	42.6	29.2	28.8	71.3	71.8	57.4	56.3
×	✓	✓	42.4	42.6	25.7	25.1	71.9	72.4	**58.3**	58.0

6.2 Experimental Results

Main Results Table 1 presents a performance comparison of different expert pruning methods on DeepSeek-V2-Lite and Qwen3-30B-A3B across general and mathematical reasoning benchmarks at expert retention ratios of r=50% and r=75%.

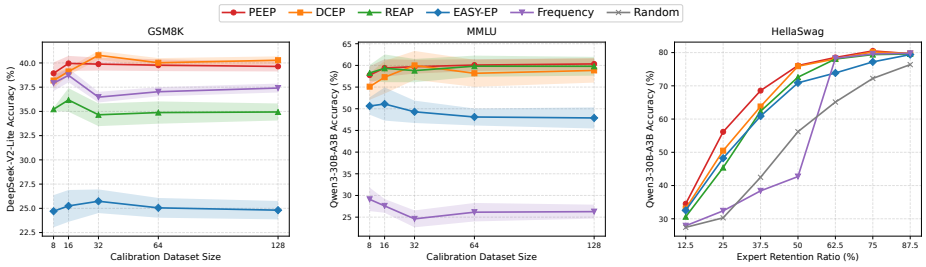

Figure 3: *Left*: Performance comparison among different pruning methods with different calibration dataset sizes on DeepSeek-V2-Lite model and GSM8K task at expert retention ratio of r=75%. *Middle*: Performance comparison among different pruning methods with different calibration dataset sizes on Qwen3-30B-A3B model and MMLU task at expert retention ratio of r=50%. *Right*: Performance comparison among different pruning methods with different expert retention ratios on Qwen3-30B-A3B model and HellaSwag task.

Overall, our structure-preserving expert pruning methods consistently achieve strong performance across both general and mathematical reasoning tasks. Notably, among all eight combinations of model, expert retention ratio, and task category, the best average performance is always achieved by either PEEP or DCEP. The advantage is particularly pronounced on mathematical reasoning tasks with Qwen3-30B-A3B. At expert retention ratio of r=75%, PEEP achieves a mathematical reasoning average performance of 78.6%, outperforming the SOTA REAP baseline by 3.5 percentage points and remaining only 1.3 percentage points below the unpruned model. At more aggressive expert retention ratio of r=50%, DCEP retains a mathematical reasoning average performance of 60.9% on Qwen3-30B-A3B, exceeding REAP by 3.6 percentage points.

Ablation Study of Router-Derived Weights Table 2 presents a performance comparison of PEEP and DCEP under different variants of router-derived weights. The complete ablation results are provided in Tables 3 and 4 in Appendix B. The ablation results show that activation frequency and mean routing score provide effective router-derived weights as priors of expert importance. We attribute this to the complementary roles of global structural modeling and local importance estimation: PEEP and DCEP capture the global structure of functional subspace and distributional geometry among experts, whereas activation frequency and mean routing score provide stable estimates of the local importance of individual basis directions and distributional modes. Although multiplying all router signals yields the best overall performance, we adopt activation frequency as the default weighting signal in our experiments, considering both effectiveness and simplicity.

Effect of Calibration Dataset Sizes Figure 3 illustrates the effect of calibration dataset sizes, with the complete results provided in Figures 4 and 5 in Appendix C. respectively. Increasing the calibration set from a very small size generally improves the reliability of pruning and reduces the variation. Using 32 calibration samples already achieves stable results, while increasing the size to 64 or 128 yields no consistent benefit. This suggests that the structural statistics exploited by our methods can be estimated reliably from only a small calibration set, making 32 samples a favorable trade-off between calibration cost and pruning quality.

Effect of Expert Retention Ratios Figure 3 illustrates the effect of expert retention ratios, with the complete results provided in Figure 6 in Appendix D. As expected, performance generally improves as more experts are retained, while aggressive pruning leads to a marked degradation. Across a wide range of expert retention ratios, our methods are consistently competitive, showing that preserving the structure of expert outputs is particularly beneficial.

7 Conclusion

We study expert pruning for MoE LLMs in a structure-preserving framework. We instantiate the structure-preserving framework with PEEP and DCEP, which maximize projection energy in the functional subspace and minimize coverage error over the distributional modes, respectively. On DeepSeek-V2-Lite and Qwen3-30B-A3B, our methods consistently achieve the best average performance on general and mathematical reasoning benchmarks at 75% and 50% expert retention ratios.

AI USE STATEMENT

In this work, we used generative AI tools for implementing method, assisting with translation. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting dataset, interpreting results. Supporting qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, vibe coding. We have reviewed all AI-assisted work. We reviewed and verified all AI-assisted content to ensure the correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We show our complete expert pruning pipeline in Figure 2 and describe the detailed algorithms for PEEP and DCEP in Algorithms 1 and 2. We provide the detailed experimental setup in Section 6 and Appendix A. We also provide the source code for reproducing the experimental results in supplementary materials.

REFERENCES

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020.

Mingyu Cao, Gen Li, Jie Ji, Jiaqi Zhang, Ajay Jaiswal, Li Shen, Xiaolong Ma, Shiwei Liu, and Lu Yin. Condense, don't just prune: Enhancing efficiency and performance in moe layer pruning. *Transactions on Machine Learning Research*, 2026.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse mixture-of-experts via hierarchical clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Tianyu Chen, Shaohan Huang, Yuan Zhou, Yuan Chen, and Furu Wei. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*, 2022.

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, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 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. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 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.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2021b.

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 *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2026.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023.

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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Lachaux, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

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

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. Reap the experts: Why pruning prevails for one-shot moe compression. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*, 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. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Enshu Liu, Junyi Zhu, Zinan Lin, Xuefei Ning, Matthew B. Blaschko, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. *arXiv preprint arXiv:2407.00945*, 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 Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Mathematical Association of America. 2023 American Invitational Mathematics Examination, 2023. American Invitational Mathematics Examination (AIME).

Mathematical Association of America. 2024 American Invitational Mathematics Examination, 2024. American Invitational Mathematics Examination (AIME).

Mathematical Association of America. 2025 American Invitational Mathematics Examination, 2025. American Invitational Mathematics Examination (AIME).

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

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

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. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2017.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. Moe-i2: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

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

A Detailed Experimental Setup

Models We conduct experiments on DeepSeek-V2-Lite and Qwen3-30B-A3B. For each model, we first perform a forward pass over a calibration dataset to probe the MoE layers and collect the mean and standard deviation of each expert's output. We then apply our proposed PEEP and DCEP methods to prune experts under two retention ratios, i.e., retaining 75% and 50% of the experts in every MoE layer. Finally, we evaluate the resulting pruned models.

Datasets We evaluate the pruned models on two categories of downstream tasks. The first category comprises general benchmarks, including HellaSwag (Zellers et al., 2019), ARC-Challenge (Clark et al., 2018), MMLU (Hendrycks et al., 2021a), and C-Eval (Huang et al., 2023). These benchmarks evaluate commonsense reasoning, scientific question answering, broad-domain knowledge, and Chinese-language understanding, respectively, providing diverse coverage of general model capabilities. For these benchmarks, we use a shared calibration dataset consisting of 32 samples drawn from C4 (Raffel et al., 2020). The second category comprises mathematical reasoning benchmarks, including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), AIME24 (Mathematical Association of America, 2024), and AIME25 (Mathematical Association of America, 2025). These benchmarks span a broad range of difficulty, from elementary multi-step reasoning to competition-level mathematical problem solving. For GSM8K and MATH500, we use 32 samples drawn from the training split of GSM8K as the calibration dataset. For AIME24 and AIME25, we use the 30 problems from AIME23 as the calibration dataset. All evaluations are conducted on four NVIDIA A100 40GB GPUs using the vLLM backend (Kwon et al., 2023) and the LM-Evaluation-Harness framework (Biderman et al., 2024).

The selection of these evaluation benchmarks enables us to assess not only the overall robustness of the pruning methods across domains and languages, but also their ability to preserve complex reasoning capabilities that may depend more strongly on coordination among experts.

Baselines We compare our methods against random pruning, frequency-based pruning, EASY-EP (Dong et al., 2025), and REAP (Lasby et al., 2026). Random pruning and frequency-based pruning serve as simple and intuitive baselines, where experts are selected randomly or according to their activation frequencies. EASY-EP and REAP represent recent SOTA methods.

B Full Ablation Results for Router-Derived Weights

Tables 3 and 4 present the complete performance comparison of PEEP and DCEP under different variants of router-derived weights. Overall, activation frequency and mean routing score provide effective priors of expert importance across different models and pruning settings. These results are consistent with Table 2. Although multiplying all router signals yields the best overall performance, we adopt activation frequency as the default weighting signal in our experiments, considering both effectiveness and simplicity.

C Effect of Calibration Dataset Sizes

Figures 4 and 5 present the complete performance comparison of different pruning methods across calibration dataset sizes at expert retention ratios of $r{=}75\%$ and $r{=}50\%$. Increasing the calibration dataset from a very small size generally improves the reliability of pruning and reduces the variation. Using 32 calibration samples already achieves stable performance, while further increasing the size to 64 or 128 yields no consistent benefit These results show that the structural statistics exploited by PEEP and DCEP can be estimated reliably from a small calibration dataset, making 32 samples a favorable trade-off between calibration cost and pruning quality.

D Effect of Expert Retention Ratios

Figure 6 presents the complete performance comparison of different pruning methods across expert retention ratios. As expected, performance generally improves as more experts are retained, while aggressive pruning leads to a marked degradation. Across a wide range of expert retention ratios, PEEP and DCEP remain consistently competitive with the baselines. These results show that preserving the functional subspace or distributional geometry of expert outputs is beneficial across different pruning budgets.

Table 3: Ablation study of router-derived weights for PEEP on DeepSeek-V2-Lite and Qwen3-30B-A3B. Columns $freq_i^{(\ell)}$, $\hat{g}_i^{(\ell)}$, and $\bar{g}_i^{(\ell)}$ indicate whether activation frequency, mean conditional routing score, and mean routing score are included in the multiplicative weight construction. "✓" indicates that the corresponding router signal is included in the weights, whereas "×" indicates that it is not included. The best and second-best results among pruned methods under each model and expert retention ratio are highlighted in bold and underlined, respectively. "–" indicates that the model fails to complete the corresponding task.

r	Router Signals			General					Mathematical Reasoning				
	$freq_i^{(\ell)}$	$\hat{g}_i^{(\ell)}$	$\bar{g}_i^{(\ell)}$	HellaSwag	ARC-Challenge	MMLU	C-Eval	Average	GSM8K	MATH500	AIME24	AIME25	Average
DeepSeek-V2-Lite													
75%	×	×	×	73.9	41.6	**47.2**	46.3	52.2	37.0	**15.8**	–	–	26.4
	✓	✓	✓	75.2	41.0	45.2	41.9	50.8	36.8	13.0	–	–	24.9
	✓	×	×	**76.1**	43.3	45.2	39.5	51.0	39.9	13.2	–	–	26.5
	×	✓	×	75.3	**44.8**	45.5	**46.6**	**53.0**	36.9	10.2	–	–	23.6
	×	×	✓	74.9	41.8	45.4	42.7	51.2	39.1	13.2	–	–	26.2
	✓	✓	×	74.9	40.9	44.0	42.9	50.7	37.6	11.6	–	–	24.6
	✓	×	✓	75.3	41.1	44.9	44.0	51.3	39.8	14.2	–	–	**27.0**
	×	✓	✓	74.8	40.3	44.4	43.8	50.8	39.3	11.6	–	–	25.5
50%	×	×	×	62.7	33.3	28.9	**30.9**	38.9	19.0	4.0	–	–	11.5
	✓	✓	✓	64.0	33.7	26.3	28.0	38.0	**23.4**	4.8	–	–	14.1
	✓	×	×	61.7	**36.8**	28.1	26.7	38.3	**23.4**	**6.2**	–	–	**14.8**
	×	✓	×	51.7	25.4	24.0	25.0	31.5	17.6	5.6	–	–	11.6
	×	×	✓	65.9	33.0	**31.2**	28.7	**39.7**	21.8	5.8	–	–	13.8
	✓	✓	×	**66.3**	35.9	28.4	26.6	39.3	21.1	5.4	–	–	13.2
	✓	×	✓	63.7	35.4	25.9	23.6	37.1	20.8	5.8	–	–	13.3
	×	✓	✓	51.5	26.0	24.2	25.0	31.7	21.8	5.6	–	–	13.7
Qwen3-30B-A3B													
75%	×	×	×	67.1	57.9	68.6	**71.3**	66.2	86.3	82.6	63.3	63.3	73.9
	✓	✓	✓	79.5	**61.1**	75.3	67.8	70.9	89.1	88.0	73.3	**70.0**	**80.1**
	✓	×	×	**80.5**	60.6	**76.0**	68.3	**71.4**	88.9	85.6	**76.7**	63.3	78.6
	×	✓	×	79.5	58.1	71.7	68.8	69.5	88.1	85.8	60.0	50.0	71.0
	×	×	✓	79.4	60.6	74.8	66.3	70.3	**89.7**	85.6	66.7	66.7	77.2
	✓	✓	×	79.3	61.0	74.8	66.6	70.4	83.6	**88.4**	60.0	56.7	72.2
	✓	×	✓	79.7	**61.1**	75.2	68.3	71.1	80.4	85.8	60.0	60.0	71.5
	×	✓	✓	79.4	60.8	75.1	67.4	70.7	80.7	88.2	66.7	56.7	73.1
50%	×	×	×	40.6	40.0	45.2	**50.7**	44.1	80.7	45.2	40.0	26.7	48.1
	✓	✓	✓	**76.7**	**47.7**	59.6	45.8	**57.5**	88.2	33.4	**63.3**	46.7	57.9
	✓	×	×	76.1	46.6	59.7	44.4	56.7	**89.7**	40.6	50.0	**50.0**	57.6
	×	✓	×	51.6	42.6	51.0	46.4	47.9	86.4	**55.0**	60.0	**50.0**	**62.8**
	×	×	✓	75.5	43.6	59.9	44.7	55.9	89.6	39.2	56.7	46.7	58.0
	✓	✓	×	75.4	43.8	**60.1**	44.9	56.0	86.4	45.0	56.7	**50.0**	59.5
	✓	×	✓	76.6	46.9	59.5	43.3	56.6	87.3	35.8	**63.3**	46.7	58.3
	×	✓	✓	75.3	44.5	59.8	45.5	56.3	85.4	46.2	**63.3**	46.7	60.4

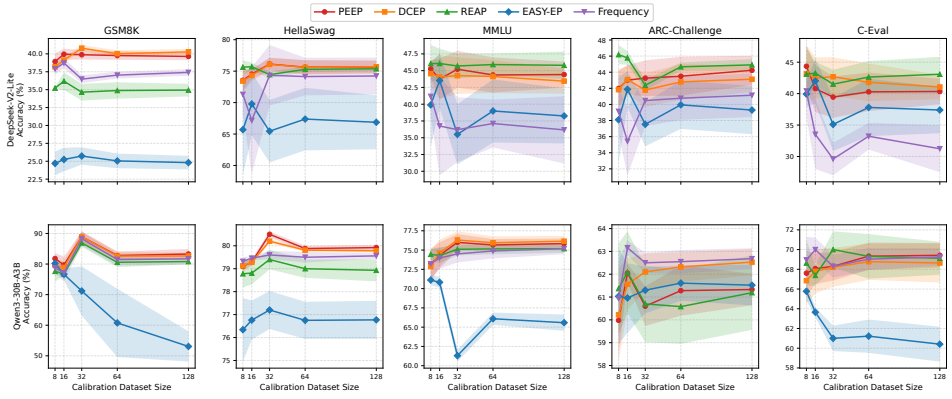

Figure 4: Performance comparison among different pruning methods with different calibration dataset sizes at expert retention ratio of r=75%. Results are averaged over five runs with calibration datasets independently sampled using different random seeds. The shaded regions represent ± one standard deviation. *Top*: DeepSeek-V2-Lite. *Bottom*: Qwen3-30B-A3B. *From left to right*: GSM8K, HellaSwag, MMLU, ARC-Challenge, and C-Eval.

Table 4: Ablation study of router-derived weights for DCEP on DeepSeek-V2-Lite and Qwen3-30B-A3B. Columns $freq_i^{(\ell)}$, $\hat{g}_i^{(\ell)}$, and $\bar{g}_i^{(\ell)}$ indicate whether activation frequency, mean conditional routing score, and mean routing score are included in the multiplicative weight construction. "✓" indicates that the corresponding router signal is included in the weights, whereas "×" indicates that it is not included. The best and second-best results among pruned methods under each model and expert retention ratio are highlighted in bold and underlined, respectively. "–" indicates that the model fails to complete the corresponding task.

r	Router Signals			General					Mathematical Reasoning				
	$freq_i^{(\ell)}$	$\hat{g}_i^{(\ell)}$	$\bar{g}_i^{(\ell)}$	HellaSwag	ARC-Challenge	MMLU	C-Eval	Average	GSM8K	MATH500	AIME24	AIME25	Average
DeepSeek-V2-Lite													
75%	×	×	×	75.4	40.4	44.6	43.7	51.0	38.5	12.8	–	–	25.7
	✓	✓	✓	75.5	43.1	**47.1**	**47.8**	**53.4**	36.8	13.0	–	–	24.9
	✓	×	×	**76.1**	41.8	44.3	42.7	51.2	**40.8**	12.6	–	–	**26.7**
	×	✓	×	75.0	**44.1**	46.2	44.7	52.5	37.4	10.6	–	–	24.0
	×	×	✓	75.0	40.4	45.3	42.6	50.8	38.4	13.0	–	–	25.7
	✓	✓	×	74.9	40.2	44.3	43.9	50.8	36.7	11.8	–	–	24.2
	✓	×	✓	75.5	40.7	44.3	43.2	50.9	38.1	**13.6**	–	–	25.8
	×	✓	✓	75.2	42.0	44.8	**45.2**	51.8	38.1	10.6	–	–	24.3
50%	×	×	×	64.1	32.3	26.3	27.0	37.4	21.4	5.2	–	–	13.3
	✓	✓	✓	63.8	**35.5**	29.1	27.9	**39.1**	21.5	4.4	–	–	12.9
	✓	×	×	62.4	34.1	26.8	25.6	37.2	21.8	**5.8**	–	–	13.8
	×	✓	×	65.0	32.6	28.2	**28.2**	38.5	16.8	**5.8**	–	–	11.3
	×	×	✓	64.5	33.2	**30.5**	27.2	38.8	**24.4**	4.4	–	–	**14.4**
	✓	✓	×	**66.2**	35.0	27.0	25.5	38.4	21.0	5.6	–	–	13.3
	✓	×	✓	64.1	32.9	24.6	22.9	36.1	23.5	5.0	–	–	14.3
	×	✓	✓	51.8	26.2	23.6	24.4	31.5	20.6	4.2	–	–	12.4
Qwen3-30B-A3B													
75%	×	×	×	69.1	56.5	69.0	68.1	65.7	88.0	82.6	**76.7**	50.0	74.3
	✓	✓	✓	79.6	61.9	75.5	**70.4**	**71.8**	**89.2**	87.0	73.3	63.3	**78.2**
	✓	×	×	**80.2**	62.1	**76.3**	68.2	71.7	88.9	86.6	70.0	60.0	76.4
	×	✓	×	76.4	57.7	72.6	69.1	69.0	87.9	85.8	63.3	**66.7**	75.9
	×	×	✓	79.4	60.8	75.1	66.5	70.4	88.4	86.4	66.7	**66.7**	77.0
	✓	✓	×	79.4	60.2	75.1	65.4	70.0	79.8	88.0	60.0	60.0	72.0
	✓	×	✓	79.6	**63.1**	75.5	66.9	71.2	76.6	86.0	66.7	60.0	72.3
	×	✓	✓	79.4	59.8	75.3	65.9	70.1	81.4	**90.4**	70.0	56.7	74.6
50%	×	×	×	45.8	33.5	47.0	**47.3**	43.4	86.9	50.6	43.3	53.3	58.5
	✓	✓	✓	76.1	**48.8**	59.8	**45.5**	**57.5**	87.3	32.2	**73.3**	43.3	59.1
	✓	×	×	75.9	45.5	60.0	44.9	56.6	**88.9**	**51.6**	50.0	53.3	60.9
	×	✓	×	58.3	37.5	52.6	43.6	48.0	85.7	47.2	56.7	56.7	**61.6**
	×	×	✓	76.0	44.7	60.4	45.1	56.6	88.1	46.0	60.0	50.0	61.0
	✓	✓	×	75.9	45.3	**60.6**	44.2	56.5	85.8	45.4	46.7	**60.0**	59.5
	✓	×	✓	**76.4**	46.1	59.1	44.2	56.4	86.4	34.6	63.3	40.0	56.1
	×	✓	✓	75.3	43.0	59.6	44.7	55.6	86.4	38.0	56.7	**60.0**	60.3

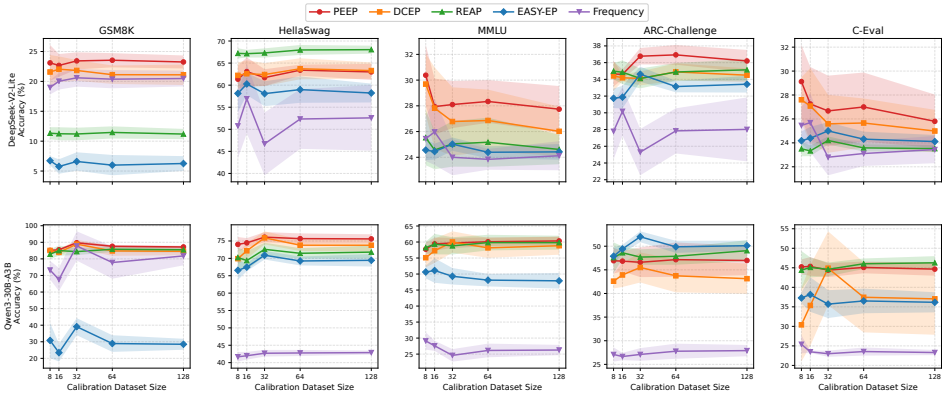

Figure 5: Performance comparison among different pruning methods with different calibration dataset sizes at expert retention ratio of r=50%. Results are averaged over five runs with calibration datasets independently sampled using different random seeds. The shaded regions represent ± one standard deviation. *Top*: DeepSeek-V2-Lite. *Bottom*: Qwen3-30B-A3B. *From left to right*: GSM8K, HellaSwag, MMLU, ARC-Challenge, and C-Eval.

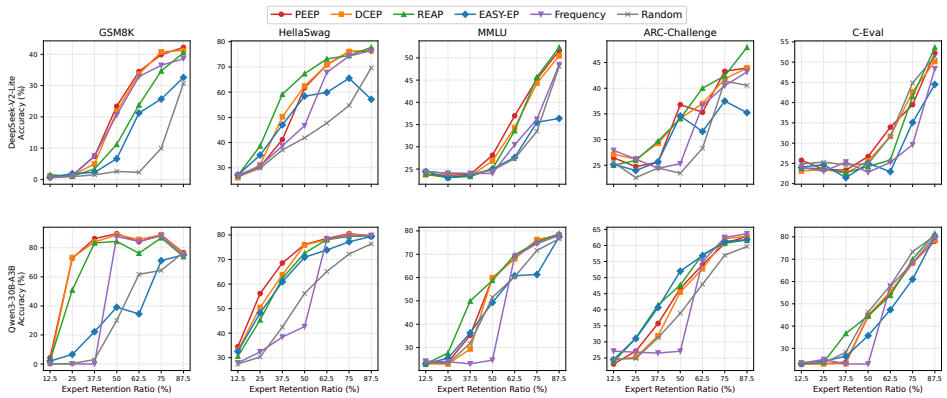

Figure 6: Performance comparison among different pruning methods with different expert retention ratios. *Top*: DeepSeek-V2-Lite. *Bottom*: Qwen3-30B-A3B. *From left to right*: GSM8K, HellaSwag, MMLU, ARC-Challenge, and C-Eval.

E SENSITIVITY TO THE SCALING FACTORS

Figure 7 presents the numbers of effective distributional modes across MoE layers under different values of the scaling factor c. We evaluate $c \in \{0.35, 0.5, 1, 1.5, 2\}$ on DeepSeek-V2-Lite and Qwen3-30B-A3B, and use c=0.5 for the results reported in Figure 1 in the main text. Increasing c reduces the estimated number of effective modes because a larger kernel bandwidth assigns higher similarity to different expert distributions. Although the absolute value varies with c, the number of effective modes remains substantially smaller than the total number of experts across all MoE layers and all tested values of c. This shows that the observed distributional redundancy is robust to the choice of the scaling factor.

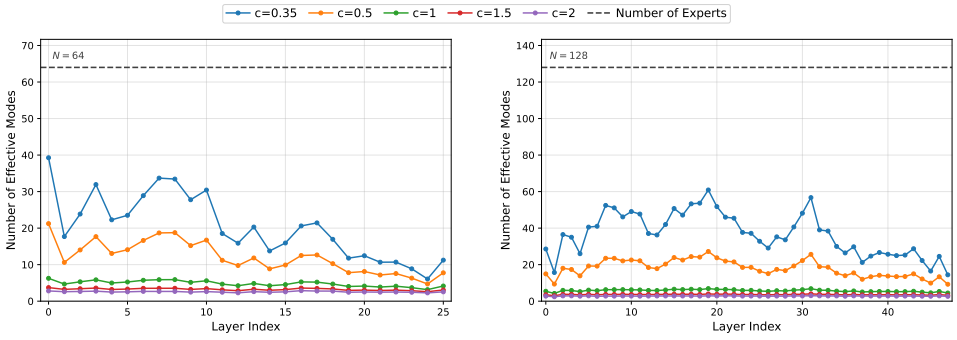

Figure 7: Numbers of effective modes of expert output distributions across MoE layers under different values of scaling factor c. *Left:* DeepSeek-V2-Lite with 64 experts per layer. *Right:* Qwen3-30B-A3B with 128 experts per layer. Across all MoE layers and different values of scaling factor c, the number of effective modes remains well below the nominal number of experts.