SLEP-MoEFICATION: SPACE-LEVEL EXPERT PARTITION FOR RICH INFORMATION PRESERVATION IN SPARSE DENSE-TO-MoE CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse dense-to-MoE conversion methods cluster pre-trained dense feed-forward networks (FFNs) into selectively activated experts to avoid the high cost of pre-training Mixture-of-Experts (MoE) models from scratch. Calibration-based approaches guide expert construction using calibration dataset to maintain performance close to the pre-trained dense model at lower inference cost. However, consistent information preservation across the input datasets is limited by reliance on calibration-dependent representations. To address this limitation, we propose a space-level expert partition method for sparse dense-to-MoE conversion (SLEP-MoEfication). Specifically, we introduce a subspace coverage score that measures the fraction of dense weight energy captured by projection onto the subspace spanned by each expert's pre-trained neuron weights. We maximize the sum of subspace coverage scores to support information preservation while keeping pre-trained weight values fixed. Our method consistently outperforms the evaluated baselines across multiple models under a common training-free centroid-based router. Our empirical analyses reveal strong positive correlations between subspace coverage and information preservation, and between information preservation and task performance. These gains hold even when random partition outperforms other existing methods. Further analysis shows that our method improves the trade-off between expert activation sparsity and performance, reduces inference cost, and improves expert load balance.

1 INTRODUCTION

Mixture-of-Experts (MoE) improves the computational efficiency of large language models (LLMs) by selectively activating a subset of experts for each token (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Zoph et al., 2022; Du et al., 2022; Jiang et al., 2024; Dai et al., 2024). However, pre-training large-scale MoE models from scratch still requires substantial data and computation (Fedus et al., 2022; Du et al., 2022). To reduce this cost, *sparse dense-to-MoE conversion* methods reuse pre-trained dense weights by assigning neurons from dense feed-forward networks (FFNs) to experts and selectively activating these experts for each token (Zhang et al., 2022b; Zuo et al., 2022; Qu et al., 2024).

A primary objective of sparse dense-to-MoE conversion is to reduce inference cost by activating only a subset of experts (Szatkowski et al., 2024; Gao et al., 2026). To this end, existing methods construct experts through weight-level clustering (Zhang et al., 2022b) or random neuron assignment (Zhu et al., 2024), while representation-level approaches use calibration data to guide expert construction and routing (Zhu et al., 2024; Gao et al., 2026; Pei et al., 2026; Zhao et al., 2026; Bamba et al., 2026). However, reported results indicate that the performance of calibration-based methods can depend on the dataset used for expert partition. These methods (Pei et al., 2026; Gao et al., 2026; Bamba et al., 2026) seek to preserve information at the representation-level and rely on calibration or training data to obtain the required activations or output targets. However, information preservation on these data does not necessarily extend to other inputs, as our experiments show substantial information loss on other task inputs for the evaluated representation-level methods (Qu et al., 2024; Pei et al., 2026) after calibration-based expert construction.

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.

To address this limitation, we examine the subspace structure of pre-trained weights to preserve information and reduce calibration dataset dependency. Our motivation comes from model merging studies that aim to preserve model responses by using task-vector subspaces as surrogates for representation structure without accessing task data (Cheng et al., 2025; Qiu et al., 2026). These studies demonstrate how parameter subspaces can guide preservation objectives without explicitly collecting representations from task data. However, their formulations constrain the integration of task-specific parameter updates. We instead investigate how the subspaces spanned by existing FFN neurons guide expert partition for information preservation without calibration data or task-specific updates.

Based on this perspective, we introduce a subspace coverage score as a data-free criterion for expert partition. The score measures the fraction of dense weight energy captured by projection onto each expert subspace. We optimize neuron assignments by maximizing the sum of these scores across experts while keeping all pretrained weight values fixed. This objective encourages individual expert subspaces to capture information from the dense weights without relying on calibration data. Our method combines the resulting partitions with a training-free centroid-based router and achieves better task performance than the evaluated partition methods.

Our main contributions are summarized as follows:

- We propose a data-free space-level expert partition method that maximizes summed subspace coverage to support information preservation.
- We provide a spectral interpretation showing that the subspace coverage objective assigns greater weight to high-energy dense weight directions within individual expert subspaces.
- Under a common training-free centroid-based router, our method improves task performance. Our analyses further show stronger high-energy subspace alignment and strong positive correlations linking subspace coverage to information preservation and information preservation to task accuracy.
- We further show that the proposed partitions improve the trade-off between expert activation sparsity and task performance, reduce inference FLOPs at matched performance targets, and improve load balance.

2 PRELIMINARIES

In this section, we describe dense FFNs with and without gating and introduce the notation for sparse dense-to-MoE conversion. Let $\boldsymbol{F}^\ell \in \mathbb{R}^{L\times D}$ denote the input projection matrix of the dense FFN in the ℓ-th layer of a pre-trained model, where L and D are the hidden and intermediate dimensions, respectively. We omit the layer index ℓ below and denote the corresponding output projection matrix by $\boldsymbol{F}' \in \mathbb{R}^{D\times L}$. We consider a row-vector input $\boldsymbol{x} \in \mathbb{R}^{1\times L}$ and omit biases for simplicity.

Non-Gated FFNs A non-gated FFN applies an elementwise activation to the input projection before mapping the intermediate representation back to the hidden dimension:

$$\boldsymbol{W}(\boldsymbol{x}) = \mathrm{act}(\boldsymbol{x}\boldsymbol{F})\boldsymbol{F}', \tag{1}$$

where $\mathrm{act}(\cdot)$ denotes the activation function.

Gated FFNs A gated FFN uses separate gate and up projections, $\boldsymbol{F}_{\mathrm{gate}}, \boldsymbol{F}_{\mathrm{up}} \in \mathbb{R}^{L\times D}$. For gated FFNs with Swish activation, the dense output is

$$\boldsymbol{W}(\boldsymbol{x}) = \left[\mathrm{Swish}(\boldsymbol{x}\boldsymbol{F}_{\mathrm{gate}}) \odot (\boldsymbol{x}\boldsymbol{F}_{\mathrm{up}})\right]\boldsymbol{F}', \tag{2}$$

where $\odot$ denotes elementwise multiplication. For these FFNs, we use $\boldsymbol{F} = \boldsymbol{F}_{\mathrm{gate}}$ throughout our expert partition formulation and spectral analysis. The corresponding columns of $\boldsymbol{F}_{\mathrm{up}}$ and rows of $\boldsymbol{F}'$ follow the neuron assignments determined from $\boldsymbol{F}$.

Sparse Dense-to-MoE Conversion Sparse conversion clusters the D intermediate neurons into E experts according to a method-specific objective. Let $\mathcal{E} = \{1, \ldots, E\}$ denote the expert index set. We assume equal-sized experts with $d = D/E$ neurons per expert (Zhang et al., 2022b; Pei et al.,

2026; Zhu et al., 2024). Let the partition set $\Pi = \{\mathcal{I}_i\}_{i\in\mathcal{E}}$ denote the full neuron partition. Each index set $\mathcal{I}_i$ contains the neurons assigned to expert i and satisfies

$$\bigcup_{i\in\mathcal{E}} \mathcal{I}_i = \{1,\dots,D\}, \qquad \mathcal{I}_i \cap \mathcal{I}_j = \varnothing \quad (i \neq j), \qquad |\mathcal{I}_i| = d \quad (i \in \mathcal{E}). \tag{3}$$

Expert i contains the projection blocks $\boldsymbol{F}_i = \boldsymbol{F}_{:,\mathcal{I}_i} \in \mathbb{R}^{L\times d}$, $\boldsymbol{F}'_i = \boldsymbol{F}'_{\mathcal{I}_i,:} \in \mathbb{R}^{d\times L}$. For gated FFNs, $\boldsymbol{F}_i$ is the gate projection block, and the expert additionally contains $\boldsymbol{F}_{\mathrm{up},i} = (\boldsymbol{F}_{\mathrm{up}})_{:,\mathcal{I}_i} \in \mathbb{R}^{L\times d}$.

A common permutation of the input projection columns and corresponding output projection rows groups neurons by expert without changing the dense FFN function. For gated FFNs, this permutation applies to both the gate and up projection columns. Under this ordering, the projection matrices have the block form $\boldsymbol{F} = [\boldsymbol{F}_1 \quad \cdots \quad \boldsymbol{F}_E]$, $\boldsymbol{F}' = \left[(\boldsymbol{F}'_1)^\top \quad \cdots \quad (\boldsymbol{F}'_E)^\top\right]^\top$. The up projection has the same column-block structure for gated FFNs.

For each input $\boldsymbol{x}$, the router selects an index set $\mathcal{S}(\boldsymbol{x}) \subseteq \mathcal{E}$ containing the k experts with the highest routing scores, so $|\mathcal{S}(\boldsymbol{x})| = k$. We define expert activation sparsity as $1 - k/E$, the fraction of experts that remain inactive for each token. For $\mathcal{S} = \mathcal{S}(\boldsymbol{x})$, we concatenate the selected input projection blocks column-wise and stack the corresponding output projection blocks row-wise in the same expert order. This gives $\boldsymbol{F}_{\mathcal{S}} \in \mathbb{R}^{L\times kd}$ and $\boldsymbol{F}'_{\mathcal{S}} \in \mathbb{R}^{kd\times L}$. The sparse FFN output is

$$\boldsymbol{y} = \boldsymbol{W}_{\mathrm{sparse}}(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{i\in\mathcal{S}(\boldsymbol{x})} \alpha_i(\boldsymbol{x})\, \mathrm{act}(\boldsymbol{x}\boldsymbol{F}_i)\boldsymbol{F}'_i, & \text{non-gated FFNs,} \\ \displaystyle\sum_{i\in\mathcal{S}(\boldsymbol{x})} \alpha_i(\boldsymbol{x})\, [\mathrm{Swish}(\boldsymbol{x}\boldsymbol{F}_i) \odot (\boldsymbol{x}\boldsymbol{F}_{\mathrm{up},i})]\, \boldsymbol{F}'_i, & \text{gated FFNs,} \end{cases} \tag{4}$$

where $\alpha_i(\boldsymbol{x})$ denotes the scalar routing weight assigned to expert i and satisfies $\alpha_i(\boldsymbol{x}) = 1$ for $i \in \mathcal{S}(\boldsymbol{x})$ and $\alpha_i(\boldsymbol{x}) = 0$ for $i \notin \mathcal{S}(\boldsymbol{x})$ and $\sum_{i=1}^{E} \alpha_i(\boldsymbol{x}) = k$ (Zhang et al., 2022b; Zhao et al., 2026). The partition set Π specifies all neuron assignments, whereas $\mathcal{S}(\boldsymbol{x})$ specifies the experts activated for a particular input.

3 Related Works

Mixture-of-Experts Training MoE models from scratch requires substantial data and computation to jointly learn expert and router parameters while maintaining training stability (Fedus et al., 2022; Du et al., 2022). Prior work supports stable training by regulating expert utilization through auxiliary losses (Shazeer et al., 2017; Fedus et al., 2022; Du et al., 2022) or routing mechanisms that directly control token-to-expert assignments (Lewis et al., 2021; Zhou et al., 2022; DeepSeek-AI et al., 2025). Although these mechanisms improve training stability, they do not reduce the substantial data and computation required for MoE pre-training. We therefore adopt sparse dense-to-MoE conversion to reuse a pre-trained dense model and avoid the substantial cost of training an MoE model from scratch.

Sparse Dense-to-MoE Conversion Previous works on sparse dense-to-MoE conversion aim to reduce inference cost while preserving dense FFNs' information. Weight-level approaches construct experts through weight clustering (Zhang et al., 2022b; Qiu et al., 2024) or random neuron assignment (Zhu et al., 2024) without calibration data. Differently, we use the dense weight energy captured by each expert subspace as a structural criterion for information preservation. Representation-level approaches use input dependent features and importance scores for expert construction (Qu et al., 2024) or expert output norms for router training (Szatkowski et al., 2024). In particular, output reconstruction and distillation objectives aim to preserve information from the dense FFN (Gao et al., 2026; Bamba et al., 2026). These preservation objectives rely on representations or output targets obtained from calibration or training data. Differently, our method formulates expert partition at the space-level and maximizes weight subspace coverage to support information preservation without calibration data.

4 Motivation: Limitations of Information Preservation

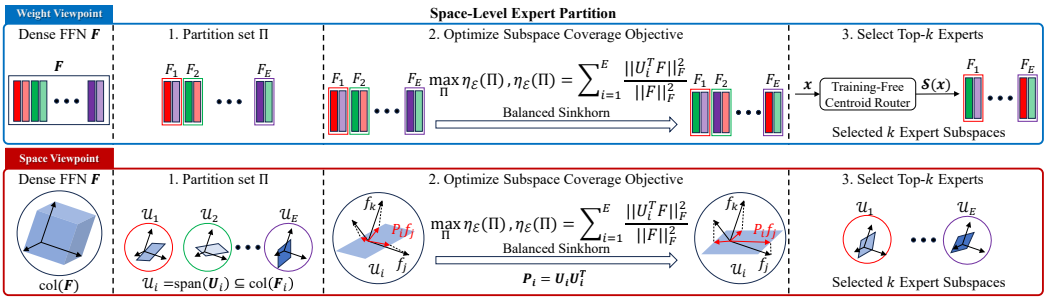


Figure 1: Overview of space-level expert partition. The blue and red boxes illustrate our process from the weight and space perspectives, respectively. We initialize the neuron partition set $\Pi = \{\mathcal{I}_i\}_{i\in\mathcal{E}}$ by randomly partitioning the columns of the dense FFN input projection $\boldsymbol{F}$ into E equal-sized groups, where $\mathcal{I}_i$ contains the neuron indices assigned to expert i. We then optimize the neuron partition set using balanced Sinkhorn (Sinkhorn, 1964; Cuturi, 2013) with the subspace coverage objective while keeping all pre-trained weight values fixed, where $\boldsymbol{f}_j$ denotes the j-th column of the dense FFN weight $\boldsymbol{F}$, and $\boldsymbol{P}_i\boldsymbol{f}_j$ is its orthogonal projection onto the expert subspace $\mathcal{U}_i$. Finally, a training-free centroid-based router selects the top-k experts for each token to perform sparse FFN.

Pei et al. (2026); Zhu et al. (2024) use activations from a calibration corpus to guide expert partition. Since these representations depend on the calibration inputs, we examine whether the resulting experts preserve information consistently on other task inputs. We then measure the Gaussian mutual information (GMI) (Cover & Thomas, 2006) change from WikiText-2 to other corpora and subtract the corresponding change for SLEP-MoEfication (ours). As shown in Figure 2, CMoE exhibits the largest negative relative changes across both models after calibration on WikiText-2. These results suggest that information preservation of baselines is sensitive to inputs beyond its calibration corpus. This observation show a limitation of relying on a calibration dataset to guide information preservation across different input distributions.

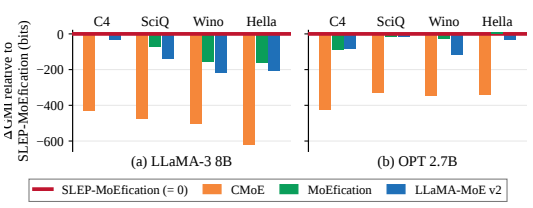


Figure 2: GMI change from WikiText-2 to other inputs, relative to Ours.

5 Method

Based on our observations in Figure 2, we propose the space-level expert partition method for data-free dense-to-MoE conversion (SLEP-MoEfication). Our method constructs experts directly from pre-trained dense FFN weights without calibration datasets. We introduce a subspace coverage score to quantify the dense FFN $\boldsymbol{F}$ weight energy captured within individual expert subspaces. We optimize neuron assignments using this criterion while keeping all pre-trained weight values fixed. We first formulate the subspace coverage objective and then provide its spectral interpretation in Sections 5.2 and 5.3.

5.1 Space-Level Expert Partition

We optimize the neuron partition set $\Pi = \{\mathcal{I}_i\}_{i\in\mathcal{E}}$ subject to the constraints in Equation (3). Each index set $\mathcal{I}_i$ determines the corresponding expert projection blocks $\boldsymbol{F}_i$ and $\boldsymbol{F}'_i$ defined in Section 2. The input projection $\boldsymbol{F}$ provides the neuron weight vectors used for expert construction. Each candidate partition assigns its columns to E experts with d neurons per expert. All parameters associated with a neuron follow the same assignment across the FFN projections. The input projection columns and their corresponding output projection ($\boldsymbol{F}'$) rows therefore remain paired. Our optimization changes expert composition without updating the numerical values of both $\boldsymbol{F}$ and $\boldsymbol{F}'$.

5.2 Subspace Coverage Objective for Space-Level Expert Partition

We aim to preserve information through the subspace structure of the pre-trained input projection. We assume $\|\boldsymbol{F}\|_F > 0$ so that its weight energy can be used for normalization. For each expert

i, we retain r_i left singular vectors associated with nonzero singular values of $\boldsymbol{F}_i$. These vectors form an orthonormal basis $\boldsymbol{U}_i \in \mathbb{R}^{L\times r_i}$, with $0 \le r_i \le \mathrm{rank}(\boldsymbol{F}_i)$. The retained expert subspace and its orthogonal projector are $\mathcal{U}_i = \mathrm{span}(\boldsymbol{U}_i)$ and $\boldsymbol{P}_i = \boldsymbol{U}_i\boldsymbol{U}_i^\top \in \mathbb{R}^{L\times L}$, respectively, with $\boldsymbol{U}_i^\top\boldsymbol{U}_i = \boldsymbol{I}_{r_i}$. Each retained basis vector belongs to the column space of $\boldsymbol{F}_i$. Since $\boldsymbol{F}_i$ contains columns of $\boldsymbol{F}$, the retained expert subspace satisfies $\mathcal{U}_i \subseteq \mathrm{col}(\boldsymbol{F}_i) \subseteq \mathrm{col}(\boldsymbol{F})$. We use $\mathcal{U}_i$ for an expert subspace and $\mathcal{S}(\boldsymbol{x})$ for the expert indices selected by the router. We define the subspace coverage score of an individual expert as

$$\eta_{\mathcal{I}_i} = \frac{\|\boldsymbol{P}_i\boldsymbol{F}\|_F^2}{\|\boldsymbol{F}\|_F^2} = \frac{\|\boldsymbol{U}_i^\top\boldsymbol{F}\|_F^2}{\|\boldsymbol{F}\|_F^2}. \tag{5}$$

The score measures the fraction of dense FFN weight energy captured by projection onto $\mathcal{U}_i$. The second expression follows from the orthonormality of $\boldsymbol{U}_i$ and avoids explicitly constructing $\boldsymbol{P}_i$. The total subspace coverage score is the sum of the individual expert scores:

$$\eta_{\mathcal{E}}(\Pi) = \sum_{i=1}^{E} \eta_{\mathcal{I}_i} = \sum_{i=1}^{E} \frac{\|\boldsymbol{U}_i^\top\boldsymbol{F}\|_F^2}{\|\boldsymbol{F}\|_F^2}. \tag{6}$$

Consequently, $0 \le \eta_{\mathcal{I}_i} \le 1$. Thus, $\eta_{\mathcal{E}}$ measures the sum of the fractions of dense FFN $\boldsymbol{F}$ energy captured within individual expert subspaces. Let $\mathcal{P}_{E,d}$ denote the set of partitions satisfying Equation (3). We formulate expert construction as

$$\max_{\Pi\in\mathcal{P}_{E,d}} \eta_{\mathcal{E}}(\Pi). \tag{7}$$

We seek neuron assignments that maximize the objective while keeping the pre-trained weights unchanged. Each candidate partition is evaluated directly from $\boldsymbol{F}$ without calibration datasets. The score serves as a weight-space criterion for information preservation, and its effect on task performance is evaluated empirically.

5.3 Spectral Interpretation of the Subspace Coverage Objective

In this section, we explain why maximizing the subspace coverage objective in Equation (7) encourages expert subspaces to capture high-energy directions of the dense FFN weights. We express the objective in the eigenbasis of $\boldsymbol{F}\boldsymbol{F}^\top$ to identify the dense weight directions emphasized by the score. Let $R = \mathrm{rank}(\boldsymbol{F})$ and write

$$\boldsymbol{F}\boldsymbol{F}^\top = \sum_{q=1}^{R} \lambda_q \mathbf{u}_q\mathbf{u}_q^\top, \qquad 0 < \lambda_R \le \cdots \le \lambda_1, \tag{8}$$

where the vectors $\mathbf{u}_q \in \mathbb{R}^{L\times 1}$ are orthonormal dense eigenvectors, and λ_q is the dense FFN $\boldsymbol{F}$ weight energy along direction $\mathbf{u}_q$. The vector $\mathbf{u}_q$ is distinct from the expert basis matrix $\boldsymbol{U}_i$. More details are in Equation (13). The partition-level coverage score has the spectral form

$$\eta_{\mathcal{E}}(\Pi) = \frac{\sum_{q=1}^{R} \lambda_q \sum_{i=1}^{E} \|\boldsymbol{U}_i^\top\mathbf{u}_q\|_2^2}{\sum_{q=1}^{R}\lambda_q}. \tag{9}$$

This identity follows by substituting Equation (8) into the numerator of each individual expert score:

$$\|\boldsymbol{U}_i^\top\boldsymbol{F}\|_F^2 = \mathrm{tr}\left(\boldsymbol{U}_i^\top\boldsymbol{F}\boldsymbol{F}^\top\boldsymbol{U}_i\right) = \sum_{q=1}^{R} \lambda_q \|\boldsymbol{U}_i^\top\mathbf{u}_q\|_2^2. \tag{10}$$

Summing over experts and using $\|\boldsymbol{F}\|_F^2 = \sum_{q=1}^{R}\lambda_q$ gives Equation (9). The term $\|\boldsymbol{U}_i^\top\mathbf{u}_q\|_2^2$ measures the coverage of dense direction $\mathbf{u}_q$ within expert i and lies between zero and one. The objective weights this coverage by λ_q and therefore emphasizes high-energy dense weight directions. More details are in Equations (17)–(29). We use this energy-weighted coverage as a structural criterion for information preservation and evaluate its effectiveness empirically in Section 6.1.

Table 1: Performance of sparse dense-to-MoE conversion methods at 25% expert activation sparsity on BoolQ, PIQA, SciQ, WinoGrande, ARC-C, and HellaSwag. C.D. indicates whether calibration data are used for expert partition, and Avg. denotes average accuracy (acc.) across the benchmarks.

Model	Method	C.D.	BoolQ	PIQA	SciQ	WinoGrande	ARC-C	HellaSwag	Avg.
LLaMA-3 8B	Dense	-	83.70	80.69	94.60	77.35	59.47	80.56	79.39
	Llama-MoE v2	✓	37.83	50.27	24.60	48.86	24.57	26.59	35.45
	CMoE	✓	53.33	60.50	71.20	55.64	26.19	45.67	52.09
	MoEfication	✗	41.01	51.58	24.80	50.51	26.37	26.51	36.80
	SLEP-MoEfication (ours)	✗	**57.83**	**69.64**	**79.50**	**58.56**	**35.84**	**59.62**	**60.16**
OPT 2.7B	Dense	-	61.87	74.76	80.60	61.33	34.30	59.73	62.10
	Llama-MoE v2	✓	55.20	73.72	76.60	60.77	33.87	57.79	59.66
	CMoE	✓	58.59	72.47	78.50	61.17	33.45	57.75	60.32
	MoEfication	✗	58.23	**73.88**	**81.00**	60.85	**34.47**	58.05	61.08
	SLEP-MoEfication (ours)	✗	**64.92**	73.18	80.90	**61.96**	33.87	**58.24**	**62.18**
Qwen-2.5 7B	Dense	-	88.13	79.98	95.20	75.30	64.33	78.47	80.24
	Llama-MoE v2	✓	66.79	67.36	90.10	56.43	38.57	56.36	62.60
	CMoE	✓	**83.43**	77.53	93.00	**69.61**	56.48	74.67	75.79
	MoEfication	✗	81.47	77.48	88.30	66.54	53.24	**74.97**	73.67
	SLEP-MoEfication (ours)	✗	83.15	**77.86**	**94.00**	68.82	**57.34**	73.86	**75.84**

5.4 Implementation of SLEP-MoEfication

We implement space-level expert partition for sparse dense-to-MoE conversion (SLEP-MoEfication) using two components. First, we optimize a learnable cost matrix to obtain balanced neuron-to-expert assignments under a normalized-weight relaxation of the objective in Equation (6). Second, we use a training-free centroid-based router for expert selection. Expert construction and router initialization use only pre-trained weights without calibration data or forward passes for collecting activations.

To group dense FFN neurons into expert blocks, we learn a cost matrix $C \in \mathbb{R}^{D\times E}$ whose balanced transport plan assigns each neuron to one expert. We adopt a balanced Sinkhorn relaxation (Sinkhorn, 1964; Cuturi, 2013; Mena et al., 2018; Bamba et al., 2026) to optimize C under a normalized-weight relaxation of the coverage objective in Equation (6), and obtain the final partition Π by an exact capacity-constrained assignment (Appendix C.1).

We use a training-free router because router training requires data and entangles routing quality with partition quality. Following MoEfication (Zhang et al., 2022b), each expert is represented by the mean of its ℓ_2-normalized neuron weight vectors, and the k experts with the highest cosine similarity to the hidden state are selected. All methods share this expert-selection rule.

6 Experimental Results

We evaluate on BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), WinoGrande (Sakaguchi et al., 2020), ARC-C (Clark et al., 2018), and HellaSwag (Zellers et al., 2019). We follow prior sparse dense-to-MoE conversion studies (Gao et al., 2026; Pei et al., 2026), which commonly use these benchmarks to assess the preservation of downstream capabilities after dense-to-MoE conversion.

For environmental settings, we convert OPT-2.7B (Zhang et al., 2022a), LLaMA-3 8B (Grattafiori et al., 2024), and Qwen2.5-7B (Yang et al., 2024) into MoE models with 320, 448, and 592 experts per FFN layer, respectively. To focus the comparison on expert partition quality, we exclude shared experts and apply the same training-free centroid-based routing rule to the compared MoE models. Further implementation and evaluation details are provided in Appendix C.3.

6.1 Model Performance on Benchmarks

Quantitative Results We evaluate the effectiveness of our space-level expert partition method on six language understanding and reasoning benchmarks. As shown in Table 1, SLEP-MoEfication achieves the highest average accuracy among the compared baselines across all three models. For LLaMA-3 8B, SLEP-MoEfication outperforms baselines on all six benchmarks and improves average accuracy by 8.07 percentage points over the strongest baseline. Also, SLEP-MoEfication on OPT 2.7B and Qwen-2.5 7B shows the best average results over six datasets.

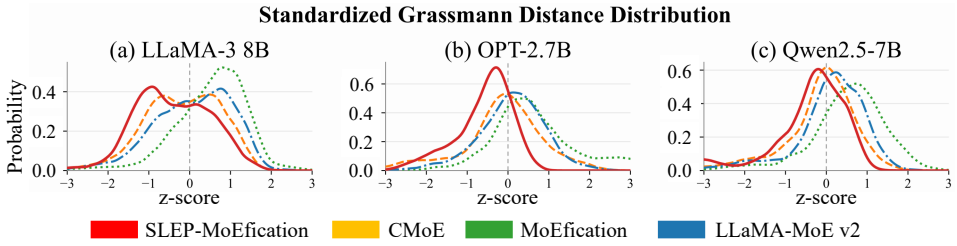


Figure 3: Standardized Grassmann distance distributions across three models. The x-axis indicates the z-score of the Grassmann distance, and lower x-axis values indicate closer alignment between expert subspaces and the leading high-energy dense subspace.

Does the Subspace Coverage Objective Capture High-Energy Directions? Our spectral interpretation in Section 5.3 shows that the subspace coverage objective assigns greater weight to higher-energy directions of the dense FFN weight matrix $\boldsymbol{F}$. To examine whether the resulting partitions align with these high-energy directions, we measure the Grassmann geodesic distance (Edelman et al., 1998) between each expert weight subspace ($\mathcal{C}_i$) and the subspace spanned by the top-d high-energy directions of $\boldsymbol{F}$ ($\mathcal{C}^{(d)}_{\text{dense}}$). More details of metrics are in Appendix D.1. Lower z-scores indicate closer alignment with the top-d high-energy dense subspace.

As shown in Figure 3, the Grassmann distance distributions of SLEP-MoEfication generally shift toward lower standardized values than those of the baselines across all three models, with the clearest separation on OPT 2.7B. This result indicates closer alignment between the expert weight subspaces and the subspace spanned by the top-d high-energy dense weight directions, consistent with our spectral interpretation.

Does Capturing High-energy Directions Preserve Information? We examine whether the subspace coverage objective supports information preservation by measuring the GMI between each expert's output and the dense FFN output. To isolate the effect of partition from routing, we evaluate each expert on task tokens without a router. For each model, we compute Pearson's correlation between the mean subspace coverage score ($\bar{\eta}_{\mathcal{I}_i}$) and mean per-expert GMI across the four partition methods. Details are in Appendix D.2. As shown in Table 2, the two measures exhibit strong positive correlations across all three models, and SLEP-MoEfication achieves the highest mean coverage and GMI. Together with the high-energy analysis in Section 6.1, these results show that greater energy-weighted coverage of dense weight directions is associated with greater information preservation, supporting our objective as a structural criterion for expert partition.

Table 2: Average per-expert subspace score $\bar{\eta}_{\mathcal{I}_i}$ ($\times 10^{-2}$) and per-expert GMI (bits, mean$_{\pm\text{std}}$ over experts), measured without a router and averaged over layers and six tasks.

Models		SLEP-MoEfication	CMoE	MoEfication	LLaMA-MoE v2	ρ_P
LLaMA-3 8B	$\bar{\eta}_{\mathcal{I}_i}\uparrow$	$\mathbf{1.606}_{\pm 0.118}$	$1.576_{\pm 0.141}$	$1.421_{\pm 0.161}$	$1.521_{\pm 0.169}$	0.97
	GMI $\uparrow$	$\mathbf{23.84}_{\pm 1.45}$	$23.76_{\pm 2.47}$	$23.62_{\pm 3.92}$	$23.73_{\pm 4.00}$	
OPT-2.7B	$\bar{\eta}_{\mathcal{I}_i}\uparrow$	$\mathbf{3.995}_{\pm 0.113}$	$3.711_{\pm 0.286}$	$3.228_{\pm 0.317}$	$3.569_{\pm 0.233}$	0.95
	GMI $\uparrow$	$\mathbf{16.71}_{\pm 2.17}$	$16.34_{\pm 5.79}$	$16.12_{\pm 6.79}$	$16.39_{\pm 6.38}$	
Qwen2.5-7B	$\bar{\eta}_{\mathcal{I}_i}\uparrow$	$\mathbf{2.211}_{\pm 0.098}$	$1.991_{\pm 0.258}$	$1.770_{\pm 0.281}$	$1.853_{\pm 0.241}$	0.92
	GMI $\uparrow$	$\mathbf{17.82}_{\pm 1.73}$	$17.69_{\pm 2.83}$	$17.45_{\pm 4.15}$	$17.65_{\pm 3.85}$	

Does the Objective Impact on Model Generalization? To assess how well the MoE layer retains the representation of the original dense FFN, we measure the GMI between the dense and MoE FFN outputs. For each method m and task t, we compute the GMI $G_{m,t}$ on the task's inputs and the accuracy with respect to the dense baseline, $A_{m,t} = \text{acc}_{m,t}/\text{acc}_{\text{dense},t}$, where $\text{acc}_{\text{dense},t}$ denotes the accuracy of the original dense FFN model. Since both differ in scale across tasks, we standardize each metric across the four methods within each task, $z_{m,t} = (x_{m,t} - \mu_t)/\sigma_t$ for $x \in \{G, A\}$, where μ_t and σ_t denote the mean and standard deviation across the methods for task t, and report the Pearson correlation ρ_P across all methods and tasks.

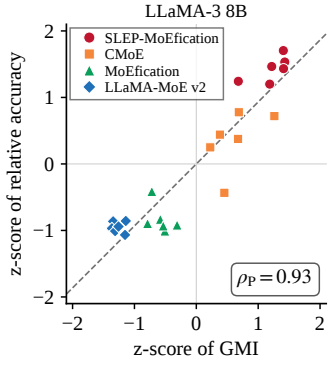


Figure 4: Correlation between GMI and acc. on LLaMA-3 8B.

On LLaMA-3 8B, GMI and relative accuracy exhibit a strong positive correlation ($\rho_P = 0.93$), as shown in Figure 4. To-

gether with the high-energy alignment in Section 6.1 and the positive coverage–GMI correlations in Section 6.1, this result supports information preservation as a possible link between high-energy weight coverage and downstream performance. These findings provide empirical support for our subspace coverage objective as a criterion for constructing experts that preserve useful information for prediction.

Meaningful Impacts on Sparse Dense-to-MoE conversion among Partition Methods Random partition is reported to provide competitive results in sparse dense-to-MoE conversion (Zhu et al., 2024). This finding motivates us to examine whether a partition criterion provides additional benefits beyond arbitrary neuron grouping. We therefore compare the evaluated partition methods against random assignment under a common training-free router.

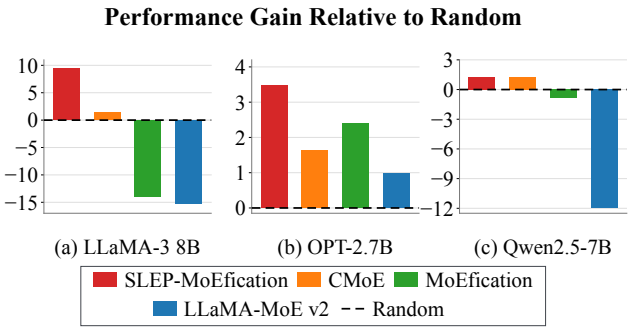


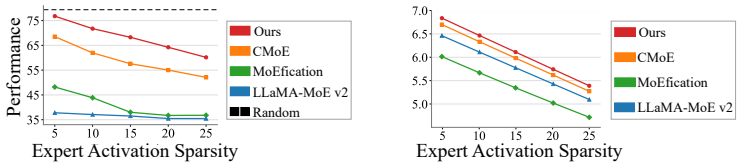
Figure 5: Difference in six-task mean accuracy relative to random partition at 25% expert activation sparsity.

As shown in Figure 5, MoEfication and LLaMA-MoE v2 improve over random partition on OPT 2.7B but underperform it on LLaMA-3 8B and Qwen-2.5 7B, indicating that their benefits depend on the backbone. Our method consistently achieves the largest performance gain over random partition across all three evaluated models at 25% expert activation sparsity. These results support subspace coverage as an effective criterion for constructing expert partitions that improve sparse inference performance beyond random assignment.

6.2 In-depth Analysis

Robustness of Information Preservation As shown in Figure 2, SLEP-MoEfication generally maintains information preservation more effectively than the evaluated baselines when inputs shift from WikiText-2 to other corpora. The predominantly negative relative GMI changes on both LLaMA-3 8B and OPT 2.7B support the robustness of our expert partitions across the evaluated input distributions. These results suggest that our partitions generally preserve information more robustly across the evaluated input distributions, while the advantage over other methods varies with the model and input corpus.

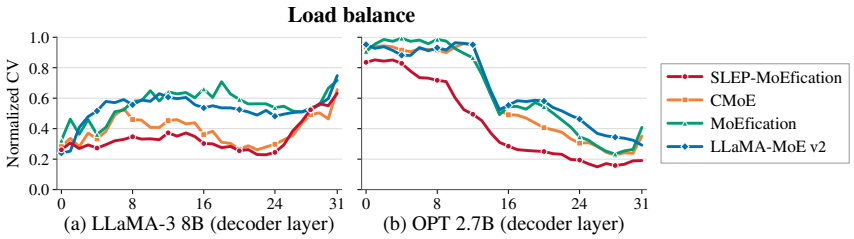
Impact on Expert Activation Sparsity We evaluate the trade-off between expert activation sparsity and task performance on LLaMA-3 8B as expert activation sparsity increases from 5% to 25% in Figure 6. To examine the coverage available under sparse activation, we sum the individual scores of the experts selected for each token, $\sum_{i \in \mathcal{S}(\boldsymbol{x}_n)} \eta_{\mathcal{I}_i}$, and average these sums over layers and 65,536 SlimPajama C4 validation tokens (Soboleva et al., 2023), where n indexes tokens and $\boldsymbol{x}_n$ denotes the input hidden state of token n at the corresponding FFN layer.



(a) Model performance. (b) Subspace coverage score.

Figure 6: LLaMA-3 8B results across expert activation sparsity levels.

For this analysis, we use each expert's full 32-dimensional subspace. As shown in Figure 6a, our method achieves the highest task performance at every evaluated sparsity level and maintains similar accuracy to CMoE at higher sparsity, requiring fewer active experts. The selected experts also achieve the highest summed coverage at each matched sparsity level in Figure 6b, indicating greater energy-weighted coverage of dense FFN directions within the same activation budget. Together,



(a) LLaMA-3 8B (decoder layer) (b) OPT 2.7B (decoder layer)

Figure 7: Load balance across the decoder layers of LLaMA-3 8B and OPT 2.7B at 25% expert activation sparsity. Lower normalized CV indicates more balanced token assignments across experts.

these results support subspace coverage as a structural criterion for constructing experts that improve the trade-off between activation sparsity and task performance.

Inference Cost Reduction To examine whether our partition method reduces the computation required for sparse inference, we compare FLOPs at matched performance targets. For each model, we evaluate configurations that reach a predefined fraction of the dense model's performance. As shown in Table 3, our method consistently reaches these targets with fewer FLOPs than the compared baselines, demonstrating an improved trade-off between task performance and inference computation.

Table 3: Relative inference FLOPs at matched performance targets. The percentage in parentheses specifies the target performance as a percentage of the corresponding dense model's performance ($\frac{\text{method acc.}}{\text{dense acc.}}$). FLOPs are evaluated when each method reaches this target and normalized to Ours ($\times 1.00$) for each model.

Model	LLaMA-MoE v2	CMoE	MoEfication	Ours
LLaMA-3 8B (75%)	$\times 1.20$	$\times 1.13$	$\times 1.20$	$\times \mathbf{1.00}$
OPT 2.7B (95%)	$\times 1.15$	$\times 1.15$	$\times 1.15$	$\times \mathbf{1.00}$
Qwen-2.5 7B (95%)	$\times 1.19$	$\times 1.05$	$\times 1.05$	$\times \mathbf{1.00}$

Impact on Load Balance In MoE architectures, uneven routing causes token concentration on a few dominant experts while leaving others underutilized, making balanced load distribution essential to fully leverage the model's total capacity. We measure how evenly each layer spreads tokens over its experts with the coefficient of variation (CV) of the expert loads (Shazeer et al., 2017). Each layer has E experts and activates k of them for every token. Let $N_{\ell,i}$ denote the total count of tokens that layer ℓ routes to expert i, and let μ_ℓ and σ_ℓ be the mean and standard deviation of $\{N_{\ell,i}\}_{i=1}^{E}$. The CV of layer ℓ and its normalized form are

$$\mathrm{CV}_\ell = \frac{\sigma_\ell}{\mu_\ell}, \qquad \widetilde{\mathrm{CV}}_\ell = \frac{\mathrm{CV}_\ell}{\sqrt{E/k - 1}}. \tag{11}$$

CV_ℓ is zero when every expert receives the same number of tokens. It reaches its maximum $\sqrt{E/k-1}$ when the same k experts serve every token, since top-k routing never selects one expert twice for a token. The normalized CV therefore lies in $[0, 1]$ and can be compared across models with different E and k. We count assignments on 65,536 tokens from the SlimPajama C4 validation split. Figure 7 shows that our method achieves the lowest normalized CV in 25 of 32 LLaMA-3 8B layers and all 32 OPT 2.7B layers, indicating more balanced token assignments across experts.

7 Conclusions

To improve information preservation in sparse dense-to-MoE conversion, we propose SLEP-MoEfication, a data-free space-level expert partition method. Our method assigns neurons to experts by maximizing the sum of subspace coverage scores without modifying pre-trained weight values. Our spectral analysis shows that this objective emphasizes high-energy dense weight directions, and our experiments confirm closer alignment between the resulting expert subspaces and the leading dense weight subspaces. Higher subspace coverage correlates positively with information preservation, which also exhibits a positive correlation with task performance. Together, these findings suggest that the high-energy coverage encouraged by our objective support downstream performance through improved information preservation. This interpretation is consistent with our method's ability to maintain task performance with fewer active experts. Additional analysis shows that the resulting partitions also improve expert load balance under training-free routing.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure:

- **Used tasks with required disclosure:** Developing theoretical or conceptual formulations, formulating mathematical claims, assisting with proof writing, providing feedback on research methodology and experimental design, supporting method implementation, and assisting with translation.
- **Unused tasks with required disclosure:** Generating synthetic datasets, providing critical components for establishing mathematical claims, proposing or refining research hypotheses, cleaning or reformatting datasets, conducting qualitative or thematic data analysis, and interpreting experimental results.

Additionally, we used generative AI tools for the following tasks with recommended disclosure:

- **Used tasks with recommended disclosure:** Creating or revising software code, assisting with drafts of manuscript text, summarizing and analyzing existing literature, brainstorming, searching for information, improving the readability of the manuscript, identifying relevant literature, formatting references, suggesting manuscript organization, and providing suggestions for titles or keywords.
- **Unused tasks with recommended disclosure:** Preparing survey or interview questions, creating or modifying scientific figures or images, selecting experimental parameters, producing research artifacts, transcribing recorded research material, and identifying new research topics or research gaps.

We have reviewed all AI-assisted work. For the AI-assisted work described above, the authors first developed the underlying research content and used generative AI as an assisting tool for the disclosed tasks. All AI-assisted outputs were subsequently reviewed and verified by the authors before being incorporated into this work.

Mathematical claims, theoretical arguments, and proofs were independently checked by the authors for correctness and consistency with the proposed formulation. AI-assisted brainstorming, literature review, information retrieval, and manuscript preparation were reviewed by the authors to ensure that they accurately reflected the intended research direction. AI-assisted descriptions of the research methodology and experiments were manually checked by the authors against the actual research and experimental settings. AI-assisted method implementation and code were also manually reviewed, executed, and tested by the authors before being used in this work.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies a method for converting pre-trained dense language models into sparse Mixture-of-Experts models. Our experiments use existing pre-trained models and publicly available benchmark datasets and do not involve new human-subject studies or the collection of new personal or sensitive data. The proposed conversion method does not address potential biases or harmful behaviors inherited from the underlying pre-trained models, and such limitations may remain after conversion. In addition, although our method aims to improve inference efficiency, the experiments still require computational resources. These considerations should be taken into account when applying the resulting models in downstream settings.

REPRODUCIBILITY STATEMENT

We support the reproducibility of our work as follows:

- **Method and implementation details.** We describe the proposed space-level expert partition method and subspace coverage objective in Section 5. Details of the partition optimization and training-free centroid-based routing are provided in Appendices C.1 and C.2.

- **Theoretical analysis.** We present the spectral interpretation of the proposed objective in Section 5.3, with the detailed derivation provided in Appendix B.
- **Experimental settings.** We report the evaluated models, datasets, routing settings, and evaluation protocol in Section 6 and Appendix C.3. The hyperparameters used for expert partition are provided in Appendix C.1, and the selection of the retained subspace rank is described in Appendix C.4.
- **Code.** We provide the implementation used for our experiments in the supplementary material, together with instructions for reproducing the reported results.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Udbhav Bamba, Arnav Chavan, Aryamaan Thakur, Steve Teig, and Deepak Gupta. DOT-MoE: Differentiable optimal transport for MoEfication. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63478.

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. doi: 10.48550/arXiv.2405.14782. URL https://arxiv.org/abs/2405.14782.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10121–10143. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/cheng25h.html.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2924–2936, 2019.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.70. URL https://aclanthology.org/2024.acl-long.70/.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5547–5569. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/du22c.html.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954. URL https://doi.org/10.1137/S0895479895290954.

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. URL http://jmlr.org/papers/v23/21-0998.html.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Shangqian Gao, Ting Hua, Reza Shirkavand, Chi-Heng Lin, Zheng Tang, Zhengao Li, Longge Yuan, Fangyi Li, Zeyu Zhang, Alireza Ganjdanesh, Qian Lou, Jie Xu, and Yen-Chang Hsu. Tomoe: Converting dense large language models to mixture-of-experts through dynamic structural pruning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=RFHq46pjb6. J2C Certification.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

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 Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/lewis21a.html.

Gonzalo E. Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Zehua Pei, Hui-Ling Zhen, Lancheng Zou, Xianzhi Yu, Wulong Liu, Sinno Jialin Pan, Mingxuan Yuan, and Bei Yu. Analytical FFN-to-MoE restructuring via activation pattern analysis. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4777–4789, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.218. URL https://aclanthology.org/2026.acl-long.218/.

Zihan Qiu, Zeyu Huang, and Jie Fu. Unlocking emergent modularity in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2638–2660, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.144. URL https://aclanthology.org/2024.naacl-long.144/.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong ZHANG, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving stability, promoting plasticity. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 128492–128518, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d0da30e312b75a3fffd9e9191f8bc1b0-Paper-Conference.pdf.

Xiaoye Qu, Daize Dong, Xuyang Hu, Tong Zhu, Weigao Sun, and Yu Cheng. LLaMA-MoE v2: Exploring sparsity of LLaMA from perspective of mixture-of-experts with post-training, 2024. URL https://arxiv.org/abs/2411.15708.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 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 *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R. Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627b token cleaned and deduplicated version of RedPajama, June 2023. URL https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama.

Filip Szatkowski, Bartosz Wójcik, Mikołaj Piórczyński, and Simone Scardapane. Exploiting activation sparsity with dense to dynamic-k mixture-of-experts conversion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 43245–43273. Curran Associates, Inc., 2024. doi: 10.52202/079017-1370. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4c2092ec0b1370cce3fb5965ab255fae-Paper-Conference.pdf.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, 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. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022a. URL https://arxiv.org/abs/2205.01068.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. MoEfication: Transformer feed-forward layers are mixtures of experts. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 877–890, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.71. URL https://aclanthology.org/2022.findings-acl.71/.

Ziyu Zhao, Tong Zhu, Xin Yu, Zhi Zhang, Tiantian Fan, Jinluan Yang, Kun Kuang, Zhongyu Wei, Fei Wu, and Yu Cheng. ExpertWeaver: Unlocking the inherent MoE in dense LLMs with GLU activation patterns. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62861.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15913–15923, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.890. URL https://aclanthology.org/2024.emnlp-main.890/.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.

Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. MoEBERT: from BERT to mixture-of-experts via importance-guided adaptation. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1610–1623, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.116. URL https://aclanthology.org/2022.naacl-main.116/.

A NOTATION

Table 4 summarizes the notation used throughout the paper. Bold uppercase letters denote matrices, bold lowercase letters denote vectors, calligraphic letters denote sets or subspaces, and $\|\cdot\|_F$ denotes the Frobenius norm.

Table 4: Summary of notation.

Symbol	Description
Dense FFN (Section 2)	
ℓ	Layer index (omitted when clear from context)
L, D	Hidden and intermediate dimensions of the FFN
$\boldsymbol{F} \in \mathbb{R}^{L\times D}$	Input projection of the dense FFN; $\boldsymbol{F} = \boldsymbol{F}_{\text{gate}}$ for gated FFNs
$\boldsymbol{F}' \in \mathbb{R}^{D\times L}$	Output projection of the dense FFN
$\boldsymbol{F}_{\text{gate}}, \boldsymbol{F}_{\text{up}}$	Gate and up projections of a gated FFN
$\boldsymbol{f}_j$	j-th column of $\boldsymbol{F}$ (weight vector of neuron j)
$\boldsymbol{x} \in \mathbb{R}^{1\times L}$	Row-vector input to the FFN
$W(\boldsymbol{x})$	Dense FFN output
$\mathrm{act}(\cdot), \mathrm{Swish}(\cdot)$	Activation functions
$\odot$	Elementwise product
Expert partition (Section 2, Section 5)	
E	Number of experts
$\mathcal{E} = \{1, \dots, E\}$	Expert index set
$d = D/E$	Number of neurons per expert
$\mathcal{I}_i \subseteq \{1, \dots, D\}$	Neuron index set assigned to expert i, $\lvert\mathcal{I}_i\rvert = d$
$\Pi = \{\mathcal{I}_i\}_{i\in\mathcal{E}}$	Neuron partition
$\mathcal{P}_{E,d}$	Set of feasible partitions satisfying Equation (3)
$\boldsymbol{F}_i = \boldsymbol{F}_{:,\mathcal{I}_i}$	Input projection block of expert i, $\boldsymbol{F}_i \in \mathbb{R}^{L\times d}$
$\boldsymbol{F}'_i = \boldsymbol{F}'_{\mathcal{I}_i,:}$	Output projection block of expert i, $\boldsymbol{F}'_i \in \mathbb{R}^{d\times L}$
$\boldsymbol{F}_{\text{up},i}$	Up projection block of expert i (gated FFNs)
Routing (Section 2, Appendix C.2)	
k	Number of experts activated per token
$1 - k/E$	Expert activation sparsity
$\mathcal{S}(\boldsymbol{x}) \subseteq \mathcal{E}$	Set of experts selected for input $\boldsymbol{x}$, $\lvert\mathcal{S}(\boldsymbol{x})\rvert = k$
$\boldsymbol{F}_{\mathcal{S}}, \boldsymbol{F}'_{\mathcal{S}}$	Concatenation of the selected input / output projection blocks
$\alpha_i(\boldsymbol{x}) \in \{0, 1\}$	Routing weight of expert i
$W_{\text{sparse}}(\boldsymbol{x}), \boldsymbol{y}$	Sparse FFN output
$\mathbf{c}_i \in \mathbb{R}^{L\times 1}$	Centroid of expert i (mean of its ℓ_2-normalized neuron weights)
$s_i(\boldsymbol{x})$	Cosine routing score of expert i
$\mathrm{TopK}(\cdot, k)$	Indices of the k largest scores
Subspace coverage objective (Section 5, Appendix B)	
r_i	Retained rank of expert i, $0 \le r_i \le \mathrm{rank}(\boldsymbol{F}_i)$; $r_i = r$ in our implementation

Symbol	Description
$U_i \in \mathbb{R}^{L \times r_i}$	Orthonormal basis of the retained left singular vectors of $\boldsymbol{F}_i$
$\mathcal{U}_i = \mathrm{span}(U_i)$	Retained expert subspace, $\mathcal{U}_i \subseteq \mathrm{col}(\boldsymbol{F}_i) \subseteq \mathrm{col}(\boldsymbol{F})$
$P_i = U_i U_i^\top$	Orthogonal projector onto $\mathcal{U}_i$
$\eta_{\mathcal{I}_i}$	Subspace coverage score of expert i, $\eta_{\mathcal{I}_i} \in [0, 1]$
$\eta_{\mathcal{E}}(\Pi)$	Total subspace coverage score, $\eta_{\mathcal{E}}(\Pi) \in [0, E]$
$\bar{\eta}_{\mathcal{I}_i}$	Average per-expert coverage score
$R = \mathrm{rank}(\boldsymbol{F})$	Rank of the dense input projection
$\lambda_q, \boldsymbol{u}_q$	q-th eigenvalue and unit eigenvector of $\boldsymbol{F}\boldsymbol{F}^\top$, $0 < \lambda_R \leq \cdots \leq \lambda_1$
δ_{qp}	Kronecker delta
$\omega_q = \lambda_q / \sum_p \lambda_p$	Normalized dense energy weight, $\sum_q \omega_q = 1$
$a_{iq}(\Pi) = \lVert U_i^\top \boldsymbol{u}_q \rVert_2^2$	Coverage of dense direction $\boldsymbol{u}_q$ by expert i
$a_q(\Pi) = \sum_i a_{iq}(\Pi)$	Summed coverage of direction $\boldsymbol{u}_q$ over experts
B_λ, Q_λ	Index set and eigenvector basis of a repeated eigenvalue λ
I_L, I_{r_i}	Identity matrices
Sinkhorn-based partition optimization (Appendix C.1)	
$\tilde{\boldsymbol{F}}$	Column-ℓ_2-normalized $\boldsymbol{F}$ used during optimization
$C \in \mathbb{R}^{D \times E}$	Learnable cost matrix
$C^{(0)}, \pi_0, s_0$	Initial cost, random balanced partition, initial cost scale
$\Gamma_\varepsilon(C) \in \mathbb{R}^{D \times E}$	Entropic balanced optimal-transport plan; Γ_{ji} is the soft assignment of neuron j to expert i
ε	Sinkhorn temperature
T	Number of Sinkhorn iterations per step
$\mathbf{1}_D, \mathbf{1}_E$	All-ones vectors
$M_i = \tilde{\boldsymbol{F}} \,\mathrm{diag}(\Gamma_{:,i}) \tilde{\boldsymbol{F}}^\top$	Soft weight covariance of expert i
$\tilde{U}_i \in \mathbb{R}^{L \times r}$	Top-r eigenvectors of M_i
$\mathcal{L}(C)$	Relaxed partition objective
$\pi^\star$	Final discrete assignment, $\mathcal{I}_i = (\pi^\star)^{-1}(i)$
Empirical analyses (Section 6, Appendix D)	
$\mathcal{C}_i = \mathrm{col}(\boldsymbol{F}_i)$	Full d-dimensional weight subspace of expert i
$\mathcal{C}_{\text{dense}}^{(d)}$	Subspace spanned by the top-d high-energy directions of $\boldsymbol{F}$
$\mathrm{Gr}(d, L)$	Grassmann manifold of d-dimensional subspaces of $\mathbb{R}^L$
Q_i, U_d	Orthonormal bases of $\mathcal{C}_i$ and $\mathcal{C}_{\text{dense}}^{(d)}$
$d_{\mathrm{Gr}}(\cdot, \cdot)$	Grassmann geodesic distance
$\sigma_j(\cdot)$	j-th largest singular value
m, t	Method index and task index
$G_{m,t}$	GMI of method m on task t
$\mathrm{acc}_{m,t}, \mathrm{acc}_{\text{dense},t}$	Accuracy of method m and of the dense model on task t
$A_{m,t} = \mathrm{acc}_{m,t} / \mathrm{acc}_{\text{dense},t}$	Relative accuracy
$z_{m,t}, \mu_t, \sigma_t$	Within-task z-score, mean, and standard deviation across methods
ρ_{P}	Pearson correlation coefficient
$n, \boldsymbol{x}_n$	Token index and input hidden state of token n
$N_{\ell,i}$	Number of tokens routed to expert i in layer ℓ
μ_ℓ, σ_ℓ	Mean and standard deviation of $\{N_{\ell,i}\}_{i=1}^E$
$\mathrm{CV}_\ell, \widetilde{\mathrm{CV}}_\ell$	Coefficient of variation of expert loads and its normalized form
Gaussian mutual information (Appendix D.2)	
N	Number of tokens
$X, Y \in \mathbb{R}^{N \times L}$	Dense and MoE FFN outputs on the same N tokens
$\bar{x}, \bar{y} \in \mathbb{R}^L$	Column means of X and Y
$v \in \mathbb{R}^L$	Column standard deviations of X
$\tilde{X}, \tilde{Y}$	Standardized outputs
$Z = [\tilde{X}\ \tilde{Y}] \in \mathbb{R}^{N \times 2L}$	Joint standardized output
ζ_n	n-th row of Z
$\Sigma, \Sigma_{XX}, \Sigma_{XY}, \Sigma_{YY}$	Joint sample covariance and its blocks

Symbol	Description
$\hat{\Sigma}$	Ledoit–Wolf shrunk covariance
$\rho, m, \kappa, \bar{\beta}$	Shrinkage intensity and its auxiliary quantities
$\gamma_1, \ldots, \gamma_L$	Canonical correlations between $\tilde{X}$ and $\tilde{Y}$ under $\hat{\Sigma}$
$\mathrm{GMI}(X;Y)$	Gaussian mutual information in bits

B Details of Spectral Interpretation of the Coverage Objective

We provide a detailed derivation of the spectral interpretation in Section 5.3. We use the individual score $\eta_{\mathcal{I}_i}$ and the summed objective $\eta_{\mathcal{E}}(\Pi)$ defined in Equations (5) and (6). As in Section 2, $\boldsymbol{F}$ denotes the input projection for a non-gated FFN and the gate projection for a gated FFN. We express the coverage objective in the eigenbasis of $\boldsymbol{F}\boldsymbol{F}^\top$ to identify the weight directions emphasized by the score. Fix a feasible partition set $\Pi \in \mathcal{P}_{E,d}$ and let $R = \mathrm{rank}(\boldsymbol{F})$. Since $\|\boldsymbol{F}\|_F > 0$, we have $R \geq 1$. The matrix $\boldsymbol{F}\boldsymbol{F}^\top$ is symmetric and positive semidefinite, so its nonzero spectral components give

$$\boldsymbol{F}\boldsymbol{F}^\top = \sum_{q=1}^{R} \lambda_q \mathbf{u}_q \mathbf{u}_q^\top, \qquad 0 < \lambda_R \leq \cdots \leq \lambda_1, \qquad \mathbf{u}_q^\top \mathbf{u}_p = \delta_{qp}. \tag{12}$$

Here, $\mathbf{u}_q \in \mathbb{R}^{L\times 1}$ is a unit dense eigenvector, and δ_{qp} is one when $q = p$ and zero otherwise. The vector $\mathbf{u}_q$ is distinct from the expert basis matrix $\boldsymbol{U}_i \in \mathbb{R}^{L\times r_i}$. The matrices $\boldsymbol{U}_i$ and $\boldsymbol{P}_i$ depend on Π through the neuron set $\mathcal{I}_i$, and we omit this dependence in their notation. For each partition, we fix the retained expert subspaces before evaluating the score. Our implementation uses a common retained rank $r_i = r$ across experts and keeps r fixed during partition. The derivation also applies to general retained ranks satisfying $0 \leq r_i \leq \mathrm{rank}(\boldsymbol{F}_i)$. The eigenvalue λ_q measures the input-weight energy along $\mathbf{u}_q$ because

$$\|\boldsymbol{F}^\top \mathbf{u}_q\|_2^2 = \mathbf{u}_q^\top \boldsymbol{F}\boldsymbol{F}^\top \mathbf{u}_q = \lambda_q \mathbf{u}_q^\top \mathbf{u}_q = \lambda_q. \tag{13}$$

We first establish the projection identities used in the coverage score. For expert i, the orthonormality condition $\boldsymbol{U}_i^\top \boldsymbol{U}_i = \boldsymbol{I}_{r_i}$ implies

$$\boldsymbol{P}_i^\top = \boldsymbol{P}_i, \qquad \boldsymbol{P}_i^2 = \boldsymbol{U}_i(\boldsymbol{U}_i^\top \boldsymbol{U}_i)\boldsymbol{U}_i^\top = \boldsymbol{U}_i \boldsymbol{U}_i^\top = \boldsymbol{P}_i. \tag{14}$$

Thus, $\boldsymbol{P}_i$ is the orthogonal projector onto $\mathcal{U}_i$. When $r_i = 0$, we interpret $\boldsymbol{U}_i$ as an empty basis and set $\boldsymbol{P}_i = \mathbf{0}$. The corresponding coverage is zero, and the following identities remain valid under this convention.

For any real matrix A, its squared Frobenius norm satisfies $\|A\|_F^2 = \mathrm{tr}(A^\top A) = \mathrm{tr}(AA^\top)$. Applying this identity and Equation (14) yields

$$\begin{aligned} \|\boldsymbol{P}_i \boldsymbol{F}\|_F^2 &= \mathrm{tr}\big((\boldsymbol{P}_i \boldsymbol{F})^\top (\boldsymbol{P}_i \boldsymbol{F})\big) \\ &= \mathrm{tr}\big(\boldsymbol{F}^\top \boldsymbol{P}_i^\top \boldsymbol{P}_i \boldsymbol{F}\big) \\ &= \mathrm{tr}\big(\boldsymbol{F}^\top \boldsymbol{P}_i \boldsymbol{F}\big) \\ &= \mathrm{tr}\big(\boldsymbol{F}^\top \boldsymbol{U}_i \boldsymbol{U}_i^\top \boldsymbol{F}\big) \\ &= \mathrm{tr}\big(\boldsymbol{U}_i^\top \boldsymbol{F}\boldsymbol{F}^\top \boldsymbol{U}_i\big) \\ &= \|\boldsymbol{U}_i^\top \boldsymbol{F}\|_F^2. \end{aligned} \tag{15}$$

The penultimate equality uses the cyclic invariance of the trace. This proves that the two numerators in Equation (5) measure the same projected weight energy.

The normalization term can also be expressed through the dense spectrum. Substituting Equation (12) gives

$$\begin{aligned}\|\boldsymbol{F}\|_F^2 &= \operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^\top) \\ &= \operatorname{tr}\left(\sum_{q=1}^{R} \lambda_q \mathbf{u}_q \mathbf{u}_q^\top\right) \\ &= \sum_{q=1}^{R} \lambda_q \operatorname{tr}(\mathbf{u}_q \mathbf{u}_q^\top) \\ &= \sum_{q=1}^{R} \lambda_q \mathbf{u}_q^\top \mathbf{u}_q \\ &= \sum_{q=1}^{R} \lambda_q.\end{aligned} \tag{16}$$

The denominator is therefore the sum of the positive dense eigenvalues and is strictly positive.

We next expand the projected energy of each expert in the same eigenbasis. By linearity of the trace,

$$\begin{aligned}\|\boldsymbol{U}_i^\top \boldsymbol{F}\|_F^2 &= \operatorname{tr}\big(\boldsymbol{U}_i^\top \boldsymbol{F}\boldsymbol{F}^\top \boldsymbol{U}_i\big) \\ &= \operatorname{tr}\left(\boldsymbol{U}_i^\top \left[\sum_{q=1}^{R} \lambda_q \mathbf{u}_q \mathbf{u}_q^\top\right] \boldsymbol{U}_i\right) \\ &= \sum_{q=1}^{R} \lambda_q \operatorname{tr}\big(\boldsymbol{U}_i^\top \mathbf{u}_q \mathbf{u}_q^\top \boldsymbol{U}_i\big) \\ &= \sum_{q=1}^{R} \lambda_q \operatorname{tr}\big((\boldsymbol{U}_i^\top \mathbf{u}_q)(\boldsymbol{U}_i^\top \mathbf{u}_q)^\top\big) \\ &= \sum_{q=1}^{R} \lambda_q \|\boldsymbol{U}_i^\top \mathbf{u}_q\|_2^2.\end{aligned} \tag{17}$$

The final equality follows from $\operatorname{tr}(aa^\top) = a^\top a = \|a\|_2^2$ for a real column vector a. Combining Equations (16) and (17) with Equation (5) gives the individual spectral score

$$\eta_{\mathcal{I}_i} = \frac{\sum_{q=1}^{R} \lambda_q \|\boldsymbol{U}_i^\top \mathbf{u}_q\|_2^2}{\sum_{q=1}^{R} \lambda_q}. \tag{18}$$

The partition-level score sums these individual scores. Substituting Equation (18) into Equation (6) and exchanging the two finite sums yields

$$\begin{aligned}\eta_{\mathcal{E}}(\Pi) &= \sum_{i=1}^{E} \eta_{\mathcal{I}_i} \\ &= \sum_{i=1}^{E} \frac{\sum_{q=1}^{R} \lambda_q \|\boldsymbol{U}_i^\top \mathbf{u}_q\|_2^2}{\sum_{q=1}^{R} \lambda_q} \\ &= \frac{\sum_{q=1}^{R} \lambda_q \sum_{i=1}^{E} \|\boldsymbol{U}_i^\top \mathbf{u}_q\|_2^2}{\sum_{q=1}^{R} \lambda_q}.\end{aligned} \tag{19}$$

This establishes the spectral form of the summed coverage objective in Equation (9).

To interpret the directional terms, define

$$a_{iq}(\Pi) = \|\boldsymbol{U}_i^\top \mathbf{u}_q\|_2^2, \qquad a_q(\Pi) = \sum_{i=1}^{E} a_{iq}(\Pi), \qquad \omega_q = \frac{\lambda_q}{\sum_{p=1}^{R} \lambda_p}. \tag{20}$$

The quantity a_{iq} equals the squared length of the projection of the unit dense direction $\mathbf{u}_q$ onto $\mathcal{U}_i$. Indeed,

$$\begin{aligned} a_{iq}(\Pi) &= \mathbf{u}_q^\top \boldsymbol{U}_i \boldsymbol{U}_i^\top \mathbf{u}_q \\ &= \mathbf{u}_q^\top \boldsymbol{P}_i \mathbf{u}_q \\ &= \mathbf{u}_q^\top \boldsymbol{P}_i^\top \boldsymbol{P}_i \mathbf{u}_q \\ &= \|\boldsymbol{P}_i \mathbf{u}_q\|_2^2. \end{aligned} \tag{21}$$

The projected component and its residual are orthogonal because

$$(\boldsymbol{P}_i \mathbf{u}_q)^\top (\boldsymbol{I}_L - \boldsymbol{P}_i)\mathbf{u}_q = \mathbf{u}_q^\top (\boldsymbol{P}_i - \boldsymbol{P}_i^2)\mathbf{u}_q = 0. \tag{22}$$

Consequently, the Pythagorean identity gives

$$1 = \|\mathbf{u}_q\|_2^2 = a_{iq}(\Pi) + \|(\boldsymbol{I}_L - \boldsymbol{P}_i)\mathbf{u}_q\|_2^2. \tag{23}$$

Both terms are nonnegative, so $0 \le a_{iq} \le 1$ and $0 \le a_q \le E$. Moreover, $\omega_q > 0$ and $\sum_{q=1}^R \omega_q = 1$. The spectral score can therefore be written as

$$\eta_{\mathcal{I}_i} = \sum_{q=1}^R \omega_q a_{iq}(\Pi), \qquad \eta_{\mathcal{E}}(\Pi) = \sum_{q=1}^R \omega_q a_q(\Pi). \tag{24}$$

Each individual score is an energy-weighted average of directional coverages, and the partition-level score applies the same dense energy weights to their sums over experts. These expressions also establish $0 \le \eta_{\mathcal{I}_i} \le 1$ and $0 \le \eta_{\mathcal{E}}(\Pi) \le E$.

The same decomposition gives an equivalent residual interpretation. Applying the orthogonal decomposition to every column of $\boldsymbol{F}$ yields

$$\|\boldsymbol{F}\|_F^2 = \|\boldsymbol{P}_i \boldsymbol{F}\|_F^2 + \|(\boldsymbol{I}_L - \boldsymbol{P}_i)\boldsymbol{F}\|_F^2. \tag{25}$$

Using Equations (23) and (24) and summing Equation (25) over experts gives

$$\begin{aligned} E - \eta_{\mathcal{E}}(\Pi) &= \sum_{i=1}^E \frac{\|(\boldsymbol{I}_L - \boldsymbol{P}_i)\boldsymbol{F}\|_F^2}{\|\boldsymbol{F}\|_F^2} \\ &= \sum_{i=1}^E \sum_{q=1}^R \omega_q \|(\boldsymbol{I}_L - \boldsymbol{P}_i)\mathbf{u}_q\|_2^2. \end{aligned} \tag{26}$$

For fixed E, maximizing the coverage objective is therefore equivalent to minimizing the sum of relative projection residuals of $\boldsymbol{F}$ over the expert subspaces. In the dense eigenbasis, this residual assigns greater weight to directions with larger eigenvalues.

The role of retained rank can also be made explicit. Since every column of $\boldsymbol{U}_i$ lies in $\mathrm{col}(\boldsymbol{F})$, projection onto the complete positive-eigenvalue space leaves $\boldsymbol{U}_i$ unchanged. Hence,

$$\begin{aligned} \sum_{q=1}^R a_{iq}(\Pi) &= \mathrm{tr}\left(\boldsymbol{U}_i^\top \left[\sum_{q=1}^R \mathbf{u}_q \mathbf{u}_q^\top \right] \boldsymbol{U}_i \right) \\ &= \mathrm{tr}(\boldsymbol{U}_i^\top \boldsymbol{U}_i) \\ &= r_i. \end{aligned} \tag{27}$$

Summing over experts yields

$$\sum_{q=1}^R a_q(\Pi) = \sum_{i=1}^E r_i = Er \quad \text{when } r_i = r \text{ for every } i. \tag{28}$$

Thus, for fixed retained ranks, the total unweighted directional coverage is fixed, and the objective distinguishes how this coverage is distributed across directions with different dense energies. In our implementation, differences in the objective do not arise from varying the retained rank across candidate partitions.

The energy preference can be stated directly for any two feasible partition sets Π and $\widetilde{\Pi}$. Since the dense weights and their eigenvalues remain fixed during partition, Equation (24) implies

$$\eta_{\mathcal{E}}(\widetilde{\Pi}) - \eta_{\mathcal{E}}(\Pi) = \sum_{q=1}^{R} \omega_q \left[a_q(\widetilde{\Pi}) - a_q(\Pi) \right]. \tag{29}$$

Thus, changes in summed directional coverage contribute in proportion to the corresponding dense energy. A direction with a larger λ_q has a larger coefficient in the objective. This establishes an energy-weighted preference for capturing dense input-weight directions within individual expert subspaces. A feasible change in neuron assignments can affect several directional terms at once, so an increase in the objective need not increase every individual coverage term.

Coverage of a direction is evaluated separately for every expert through a_{iq}. The same $\mathbf{u}_q$ can therefore contribute through multiple experts when their subspaces overlap. Disjoint neuron assignments constrain the column indices assigned to experts, but they do not require the resulting subspaces to be orthogonal. This explains how distinct experts can capture common dense directions without sharing neuron parameters. Because the score sums individual coverages, a direction can contribute to multiple terms, and $\eta_{\mathcal{E}}(\Pi)$ lies in $[0, E]$. The score therefore measures summed coverage across individual expert subspaces rather than the energy captured by projection onto their combined subspace.

Finally, the spectral expression does not require distinct positive eigenvalues. For a positive eigenvalue λ, let $\mathcal{B}_\lambda = \{q : \lambda_q = \lambda\}$ and let $\mathbf{Q}_\lambda \in \mathbb{R}^{L \times |\mathcal{B}_\lambda|}$ contain the corresponding orthonormal dense eigenvectors. Their combined contribution to expert i is

$$\sum_{q \in \mathcal{B}_\lambda} \lambda_q a_{iq}(\Pi) = \lambda \|\boldsymbol{U}_i^\top \mathbf{Q}_\lambda\|_F^2 = \lambda \operatorname{tr}\big(\boldsymbol{P}_i \mathbf{Q}_\lambda \mathbf{Q}_\lambda^\top\big). \tag{30}$$

The matrix $\mathbf{Q}_\lambda \mathbf{Q}_\lambda^\top$ is the projector onto the corresponding dense eigenspace and is unchanged by an orthogonal change of basis within that eigenspace. Thus, individual directional terms may depend on the eigenbasis chosen for a repeated eigenvalue, but their total weighted contribution and the coverage score remain invariant. The score is likewise unchanged by an orthogonal change of basis within a fixed retained expert subspace because such a change leaves $\boldsymbol{P}_i$ unchanged. This invariance does not extend to selecting different retained subspaces when an expert has tied singular values at the truncation boundary. In that case, a fixed subspace selection rule is needed to make the score a single-valued function of the partition.

These identities characterize the energy-weighted geometric preference of the coverage objective using only $\boldsymbol{F}$ and the retained expert subspaces. They do not establish a guarantee for nonlinear FFN output preservation, task performance, activation sparsity, or routing balance. The relationships between the coverage objective and these properties are evaluated empirically.

C Implementation and Evaluation Details

C.1 Expert Partition

For each FFN layer, let $\tilde{\boldsymbol{F}} \in \mathbb{R}^{L \times D}$ be the gate projection (`fc1` for OPT), whose columns (neurons) are ℓ_2-normalized. We split the D neurons into E experts of equal size $d = D/E$. We do not search over discrete partitions directly. Instead, we learn a cost matrix $C \in \mathbb{R}^{D \times E}$ and relax the partition to a balanced transport plan. The procedure uses only the weights and needs no calibration data. Each layer is optimized independently.

Initialization $C^{(0)} = -s_0 \cdot \mathrm{onehot}(\pi_0)$, where $\pi_0 : \{1, \dots, D\} \to \mathcal{E}$ is a random balanced assignment, $\mathrm{onehot}(\pi_0) \in \{0,1\}^{D \times E}$ has a one at entry $(j, \pi_0(j))$ and zeros elsewhere, and s_0 is the initial scale.

Soft assignment At each step, C is mapped to the entropic balanced optimal-transport plan

$$\Gamma_\varepsilon(C) = \arg\min_{\Gamma \geq 0} \langle C, \Gamma \rangle + \varepsilon \sum_{j,i} \Gamma_{ji}(\log \Gamma_{ji} - 1) \quad \text{s.t.} \quad \Gamma \mathbf{1}_E = \mathbf{1}_D,\ \Gamma^\top \mathbf{1}_D = d\, \mathbf{1}_E, \tag{31}$$

computed with T log-domain Sinkhorn iterations. The marginal constraints enforce equal expert sizes directly, so no balance penalty is needed.

Objective For each expert i, we form the soft covariance $M_i = \tilde{\boldsymbol{F}} \operatorname{diag}(\Gamma_{:,i})\, \tilde{\boldsymbol{F}}^\top$ and take its top-r eigenvectors $\tilde{U}_i \in \mathbb{R}^{L\times r}$ with a differentiable randomized subspace iteration. We minimize

$$\mathcal{L}(C) = -\sum_{i=1}^{E} \frac{\|\tilde{U}_i^\top \tilde{\boldsymbol{F}}\|_F^2}{\|\tilde{\boldsymbol{F}}\|_F^2}, \tag{32}$$

which maximizes the summed coverage of $\tilde{\boldsymbol{F}}$'s energy across individual expert subspaces, using a normalized-weight relaxation of the objective in Equation (6).

Optimization C is updated with Adam. The temperature ε follows a piecewise-constant schedule and is annealed from a near-uniform plan to a near-hard one. Each value is held for an equal share of the steps. Near a hard plan, the eigenvector gradient can become non-finite when eigenvalues are nearly degenerate; such updates are skipped.

Discretization The final partition is read off the learned cost by solving the exact capacity-constrained assignment $\pi^\star = \arg\min_\pi \sum_{j=1}^{D} C_{j,\pi(j)}$ subject to $|\pi^{-1}(i)| = d$ for all i, formulated as a min-cost flow problem. We set $\mathcal{I}_i = (\pi^\star)^{-1}(i)$ for all $i \in \mathcal{E}$, which gives the final partition Π.

Hyperparameter	Value
Neurons per expert d	32
#experts E	592 (Qwen2.5-7B), 448 (LLaMA-3-8B), 320 (OPT-2.7B)
Subspace rank r	27 (Qwen2.5-7B), 26 (LLaMA-3-8B), 26 (OPT-2.7B)
Optimizer / learning rate	Adam / 0.1
Optimization steps	150 per layer
Temperature schedule ε	$\{1.0, 0.3, 0.1, 0.03\}$, equal shares
Sinkhorn iterations T	200 per step
Initial cost scale s_0	1.0
Initialization	random balanced partition
Column normalization of $\boldsymbol{F}$	yes

Table 5: Hyperparameters of the Sinkhorn-based partition optimization.

C.2 Training-Free Centroid-Based Routing

We use an expert-centroid router following the similarity selection scheme of Zhang et al. (2022b). As in section 2, $\boldsymbol{F}$ denotes the input projection for non-gated FFNs and the gate projection for gated FFNs. For expert $i \in \mathcal{E}$, we average the normalized weight vectors of its assigned neurons:

$$\mathbf{c}_i = \frac{1}{d} \sum_{j\in\mathcal{I}_i} \frac{\boldsymbol{F}_{:,j}}{\|\boldsymbol{F}_{:,j}\|_2} \in \mathbb{R}^{L\times 1}. \tag{33}$$

The centroid summarizes the neuron weight directions within the expert. For the row-vector input $\boldsymbol{x} \in \mathbb{R}^{1\times L}$, the routing score is

$$s_i(\boldsymbol{x}) = \frac{\boldsymbol{x}\mathbf{c}_i}{\|\boldsymbol{x}\|_2 \|\mathbf{c}_i\|_2}. \tag{34}$$

Cosine similarity compares input and centroid directions independently of their norms. The router selects

$$\mathcal{S}(\boldsymbol{x}) = \operatorname{TopK}\left(\{s_i(\boldsymbol{x})\}_{i\in\mathcal{E}}, k\right), \tag{35}$$

where TopK returns the indices of the k largest scores. Selected expert outputs are summed with unit weights. We construct the router from pre-trained neuron weights without additional data or router training.

C.3 Model and Evaluation Settings

We set $r_i = r$ for all experts, with $r = 26$ for OPT and LLaMA and $r = 27$ for Qwen. For the expert activation sparsity and load-balance analyses in Figure 6 and Figure 7, we use 65,536 tokens from the SlimPajama C4 validation split (Soboleva et al., 2023). We run OPT on four RTX 3090 GPUs and LLaMA and Qwen on four RTX A6000 GPUs.

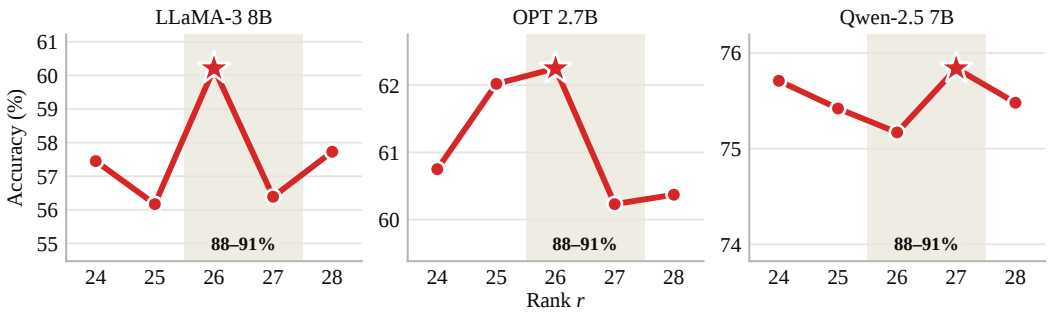


Figure 8: Effect of the subspace rank r. Average accuracy over six tasks (BoolQ, PIQA, SciQ, WinoGrande, ARC-C, HellaSwag) as r varies.

Task evaluation We use the LM Evaluation Harness (Gao et al., 2024). Following the few-shot settings of prior dense-to-MoE evaluations (Gao et al., 2026; Qu et al., 2024), few-shot counts are 32, 0, 0, 5, 25, and 10 for BoolQ, PIQA, SciQ, WinoGrande, ARC-C, and HellaSwag, respectively. We retain the Harness's `acc` metric for BoolQ and WinoGrande. For the remaining tasks, we use `acc_norm` to reduce the preference for shorter answer choices (Biderman et al., 2024). This divides each answer's log-likelihood by its character count. We report the unweighted mean of the six task accuracies as a percentage, giving each benchmark equal weight.

C.4 Selection of the Subspace Rank

We use a common retained rank $r_i = r$ across experts in Section 5.2, where $1 \le r \le \min_i \operatorname{rank}(\boldsymbol{F}_i)$. Since each expert contains $d = D/E$ neurons, the admissible rank range can vary with the expert configuration. To narrow the search, we examine the relationship between the retained rank, dense-weight cumulative energy, and downstream performance. As shown in Figure 8, the highest average accuracy over six tasks among the evaluated ranks is achieved at $r = 26$ for LLaMA-3 8B and OPT 2.7B, and at $r = 27$ for Qwen-2.5 7B. These ranks fall within the shaded interval corresponding to cumulative energy ratios of 88–91%, providing an empirical basis for narrowing the rank search.

D Experimental Analyses

D.1 Grassmann Distance

Specifically, for expert i, we define its weight subspace as

$$\mathcal{C}_i = \operatorname{col}(\boldsymbol{F}_i) \in \operatorname{Gr}(d, L), \qquad \mathcal{C}^{(d)}_{\text{dense}} = \operatorname{span}(u_1, \ldots, u_d) \in \operatorname{Gr}(d, L), \tag{36}$$

where $\operatorname{Gr}(d, L)$ denotes the Grassmann manifold of d-dimensional subspaces in $\mathbb{R}^L$ (Absil et al., 2008), and $u_1, \ldots, u_d$ are orthonormal eigenvectors corresponding to the d largest eigenvalues of $\boldsymbol{F}\boldsymbol{F}^\top$. We assume that $\boldsymbol{F}_i$ has full column rank. Let Q_i and $U_d = [u_1, \ldots, u_d]$ be orthonormal bases of the two subspaces. We compute their Grassmann geodesic distance (Edelman et al., 1998) as

$$d_{\text{Gr}}\left(\mathcal{C}_i, \mathcal{C}^{(d)}_{\text{dense}}\right) = \sqrt{\sum_{j=1}^{d} \left[\arccos\left(\sigma_j(Q_i^\top U_d)\right)\right]^2}, \tag{37}$$

where $\sigma_j(Q_i^\top U_d)$ denotes the j-th largest singular value of $Q_i^\top U_d$.

We set $d = 32$ to match the full dimension of each expert subspace. We standardize the Grassmann distances separately for each model using the mean and standard deviation pooled across all methods, layers, and experts. We compute Gaussian kernel density estimates for each method and layer using a shared bandwidth within each model and average the per-layer densities with equal layer weights. Lower z-scores indicate closer alignment with the top-d high-energy dense subspace.

D.2 Gaussian Mutual Information

We quantify how much of the dense FFN output survives the conversion with Gaussian mutual information (GMI). For an expert layer we collect the FFN output of the dense model and that of the MoE model on the same N tokens and write them as $X, Y \in \mathbb{R}^{N\times L}$, where L is the hidden size. Each model runs its own forward pass, so Y also carries the error accumulated in the layers before.

Why a Gaussian estimate The exact mutual information between the two outputs cannot be measured reliably at this scale. The outputs have $L = 2{,}560$ to $4{,}096$ dimensions and we observe $N = 65{,}536$ tokens, which is far too few for nonparametric estimators such as nearest-neighbor methods, whose bias grows rapidly with dimension. With at most N contrastive candidates, the InfoNCE lower bound cannot exceed $\log_2 N = 16$ bits, while the values we observe are in the thousands of bits. The Gaussian estimate depends only on the joint covariance, has a closed form, and is stable at this dimension once the covariance is shrunk. With the true covariance, it equals the exact mutual information when the two outputs are jointly Gaussian. In general it measures the information carried by the linear dependence between them, which the linear maps of the following layers can read out directly. We use it as a common yardstick across methods rather than as an estimate of the exact information content.

Standardization Let $\bar{x}, \bar{y} \in \mathbb{R}^L$ be the column means of X and Y, and let $v \in \mathbb{R}^L$ hold the standard deviations of the columns of X. Both outputs are centered and divided by the dense statistics,

$$\tilde{X} = (X - \mathbf{1}\bar{x}^\top)\,\mathrm{diag}(v)^{-1}, \qquad \tilde{Y} = (Y - \mathbf{1}\bar{y}^\top)\,\mathrm{diag}(v)^{-1}. \tag{38}$$

Using the dense scale for both outputs keeps a few high-magnitude channels from dominating the estimate and places the two outputs on a common scale.

Shrunk joint covariance Let $Z = [\tilde{X}\ \tilde{Y}] \in \mathbb{R}^{N\times 2L}$, let ζ_n be its n-th row, and let $\Sigma = Z^\top Z/N$ be the joint sample covariance with blocks Σ_{XX}, Σ_{XY} and Σ_{YY}. FFN outputs occupy few directions of the hidden space, so Σ is close to singular and its log-determinant is unstable. We therefore shrink it toward a scaled identity with the Ledoit–Wolf estimator (Ledoit & Wolf, 2004),

$$\hat{\Sigma} = (1-\rho)\,\Sigma + \rho\, m I_{2L}, \qquad m = \frac{\mathrm{tr}\,\Sigma}{2L}, \qquad \rho = \min\!\left(1, \frac{\bar{\beta}}{\kappa}\right), \tag{39}$$

where $\kappa = \|\Sigma - mI_{2L}\|_F^2/2L$ and $\bar{\beta} = \frac{1}{2LN^2}\sum_{n=1}^{N}\|\zeta_n^\top\zeta_n - \Sigma\|_F^2$. Because ρ depends on N, every measurement uses the same number of tokens.

Gaussian Mutual Information The GMI is the mutual information of two jointly Gaussian vectors with covariance $\hat{\Sigma}$, measured in bits,

$$\mathrm{GMI}(X;Y) = \frac{1}{2\ln 2}\left(\ln\det\hat{\Sigma}_{XX} + \ln\det\hat{\Sigma}_{YY} - \ln\det\hat{\Sigma}\right) = -\frac{1}{2}\sum_{j=1}^{L}\log_2\!\left(1-\gamma_j^2\right), \tag{40}$$

where $\gamma_1, \ldots, \gamma_L$ are the canonical correlations between the two outputs under $\hat{\Sigma}$, the singular values of $\hat{\Sigma}_{XX}^{-1/2}\hat{\Sigma}_{XY}\hat{\Sigma}_{YY}^{-1/2}$. Every direction in which the MoE output linearly predicts the dense output adds $-\frac{1}{2}\log_2(1-\gamma_j^2)$ bits, so the GMI grows both with how well and with how many directions the MoE layer preserves. We compute it for every expert layer and report the mean over layers.

Inputs We measure the GMI on three kinds of input, each with exactly $N = 65{,}536$ tokens. WikiText-2 uses 32 windows of 2,048 tokens cut from the concatenated validation text. C4 uses 32 windows of 2,048 tokens drawn from documents of the SlimPajama C4 validation split, the same sample as the load-balance measurement. For the six downstream tasks we feed the requests that lm-evaluation-harness sends to the model, with the few-shot settings of our evaluation (32 for BoolQ, 25 for ARC-Challenge, 10 for HellaSwag, 5 for WinoGrande and none for PIQA and SciQ). Each document contributes one request built from its correct answer. Documents are shuffled with a fixed seed and added until the budget is reached, and the last request is truncated to meet it exactly. All moments are accumulated in 64-bit floating point.

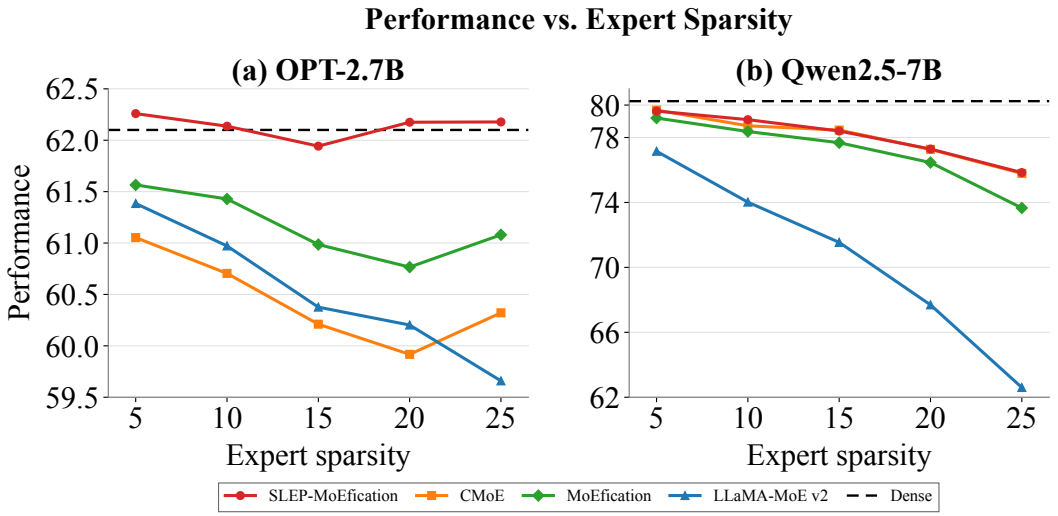


Figure 9: Mean accuracy across six tasks for OPT-2.7B and Qwen2.5-7B at varying expert activation sparsity levels. Dashed lines indicate the corresponding dense model performance.

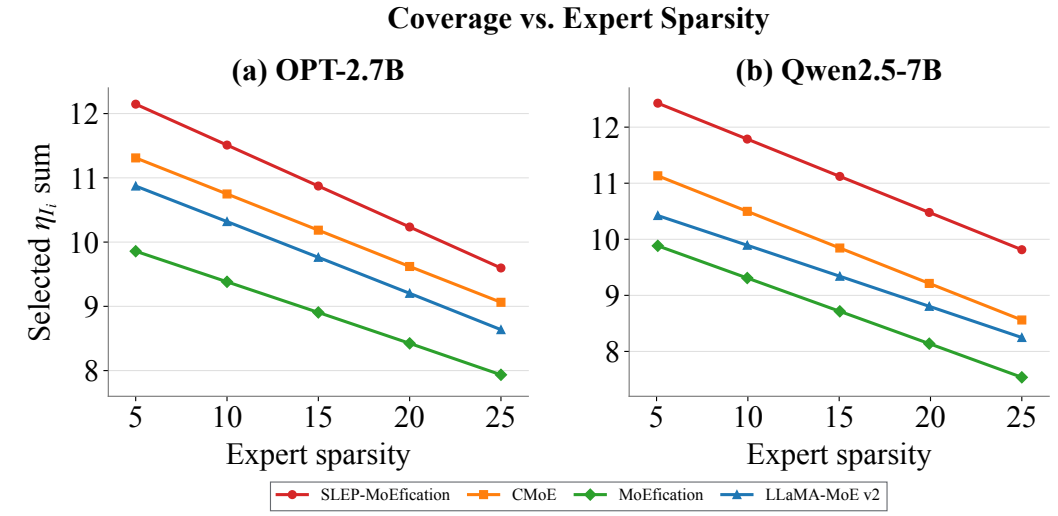


Figure 10: Summed subspace coverage of selected experts in OPT-2.7B and Qwen2.5-7B at varying expert activation sparsity levels. Values are averaged over tokens and layers.

D.3 Additional Results on Expert Activation Sparsity

Settings We extend the LLaMA analysis in Figure 6 to OPT-2.7B and Qwen2.5-7B. We use $k \in \{240, 256, 272, 288, 304\}$ for OPT and $k \in \{444, 474, 503, 533, 562\}$ for Qwen. With 320 and 592 experts per layer, respectively, these settings cover approximately 5%–25% expert activation sparsity.

Selected subspace coverage We use 32 document-local windows of 2,048 tokens from the SlimPajama C4 validation split (Soboleva et al., 2023), tokenized with each model's tokenizer. The same windows are used across methods and sparsity levels within each model. We evaluate η_{I_i} in Equation (5) using each expert's full 32-dimensional subspace. For each token, we sum the scores of its selected experts and average these sums over tokens and all FFN layers.

Results and analysis Figure 9 shows that our method achieves the highest mean accuracy at every evaluated sparsity level on OPT. Its accuracy ranges from 61.94% to 62.26%, compared with 62.10% for the dense model. On Qwen, its accuracy is close to CMoE, which is slightly higher at approximately 5% and 15% sparsity. At 25% sparsity, our method achieves 75.84% and CMoE

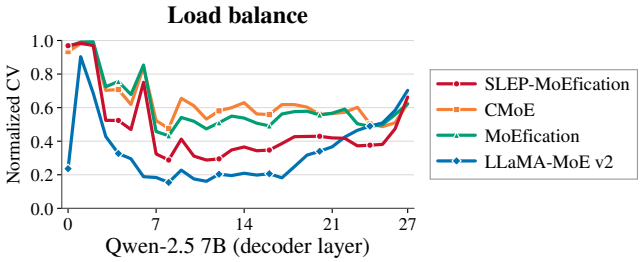


Figure 11: Load balance across the decoder layers of Qwen-2.5 7B at 25% expert activation sparsity.

achieves 75.79%. Our method outperforms MoEfication and LLaMA-MoE v2 throughout the evaluated range on both models.

Figure 10 shows that our method achieves the highest selected coverage sum at every evaluated sparsity level on both models. As sparsity increases from approximately 5% to 25%, its sum decreases from 12.15 to 9.60 on OPT and from 12.43 to 9.81 on Qwen. Although all methods show decreasing sums as fewer experts are activated, our method consistently provides greater summed coverage of dense FFN weight energy under the same activation budget.

D.4 Load Balance on Qwen-2.5 7B

Figure 11 extends the analysis of Figure 7 to Qwen-2.5 7B. LLaMA-MoE v2 spreads tokens evenly because its router has a weaker fixed preference among experts. All four methods route with the same rule and select the 444 experts whose centers have the highest cosine similarity with the hidden state. We measure this fixed preference as the share of a token's score variance that is a fixed per-expert offset, using the dense model's hidden states for 16,384 C4 validation tokens. The share is 0.27 for LLaMA-MoE v2 against 0.41 for SLEP-MoEfication and 0.61 to 0.62 for CMoE and MoEfication. With a weaker fixed preference, the selected experts change from token to token and the load evens out. However, an even load alone does not guarantee that the router picks the right experts. Given that LLaMA-MoE v2 achieves the lowest accuracy among the four methods, its selection is likely driven by centers that carry little information about what each expert computes. The low CV of LLaMA-MoE v2 therefore does not necessarily imply efficient use of its experts.