FA-MoE: Fisher-Adaptive Routing for Sparse Vision-Language Mixture-of-Experts

Anonymous authors
Paper under double-blind review

Abstract

Sparse Mixture-of-Experts (MoE) architectures have emerged as an efficient paradigm for scaling large vision-language models (LVLMs). In dense-to-MoE upcycling, experts inherit weights from a pretrained dense model, while the newly introduced router is commonly initialized randomly and trained with an auxiliary load-balancing objective. This leaves two gaps. The first arises at initialization: the experts inherit pretrained structure, whereas the router starts without information about the representation space it must partition. The second arises during training: auxiliary load balancing controls aggregate expert usage but does not directly regulate each token's routing distribution. We introduce Fisher-Adaptive MoE (FA-MoE) to address both gaps. FA-MoE initializes the router with Fisher-discriminative directions derived from the dense model's hidden representations, using K-means cluster assignments as pseudo-labels. During training, it combines token-level routing regularization that (1) controls probability mass outside the selected top-k experts and (2) adapts entropy within the selected set according to routing confidence, with batch-level regularization that maintains balanced expert utilization. Across StableLM-1.6B, Qwen-1.8B, and Phi-2 (2.7B) on eight multimodal benchmarks, FA-MoE ranks first in 21 of 24 backbone–benchmark settings. Controlled ablations show that Fisher initialization performs better than random initialization under the full routing objective, while routing analyses show fewer subsequent changes in the selected expert set.

1 Introduction

Sparse Mixture-of-Experts (MoE) architectures scale model capacity efficiently by routing each token to a small subset of k expert networks rather than activating all experts (Shazeer et al., 2017; Fedus et al., 2022; Dai et al., 2024; Jiang et al., 2024). A standard MoE layer has two components: a router that scores each token against every expert and selects the top-k, and a set of experts, typically feed-forward networks (FFNs). Recent work has extended sparse MoE to large vision-language models (LVLMs), where conditional computation is especially attractive given the high cost of multimodal training and inference (Lin et al., 2026; Li et al., 2024b).

Sparse LVLMs can be constructed by *upcycling* a pretrained dense model: expert weights are initialized by copying the dense FFN weights (Komatsuzaki et al., 2023; He et al., 2024), while the router is initialized independently, as in MoE-LLaVA (Lin et al., 2026). This creates an initialization asymmetry: the experts inherit pretrained weights, whereas the router starts without information about the representation space it must partition. Moreover, the copied experts are functionally identical at initialization, so early routing determines which token updates each expert copy receives as sparse training begins. Because these experts initially produce similar outputs, the task loss provides only limited early feedback for learning distinct routing preferences.

Our key observation is that the pretrained dense model already contains representation structure that can provide a useful prior for routing. As shown in Figure 1, random initialization produces interleaved expert assignments within the same representation regions, whereas Fisher initialization better follows the local structure already present in the dense representation space. Motivated by this observation, we cluster dense hidden representations with K-means and use the resulting assignments only as pseudo-labels to compute Fisher-discriminative directions for router initialization. Unlike centroid-based initialization, these directions account for both between-group separation and within-

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

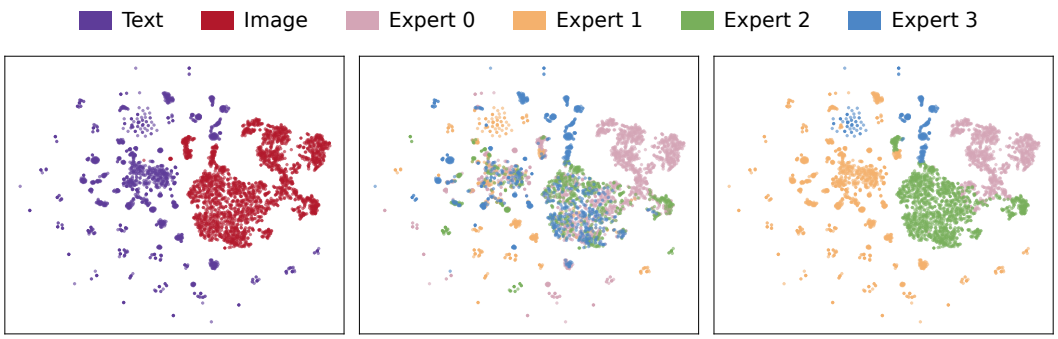

Figure 1: Router assignments at initialization on held-out tokens from MoE layer 2 of StableLM-1.6B. All panels show the same t-SNE projection of the token representations; in (b) and (c), each token is colored by its top-1 expert, the expert with the highest router score. (a) Tokens colored by modality for reference: the representations form separated regions with finer clusters. (b) With random initialization, different experts are interleaved within the same regions. (c) With Fisher initialization, tokens within the same local regions more often share an expert, so the expert assignments better follow the separations visible in (a). This suggests that the geometry already present in the dense representation space can provide a useful prior for router initialization.

group variation. This gives the router an initialization derived from the pretrained representation space rather than an independent random start.

Good initialization alone does not guarantee confident routing throughout training. Standard load-balancing objectives regulate expert utilization at the batch level, but do not constrain how strongly an individual token commits to its selected top-k experts. In a top-k layer, the router produces a probability distribution over all experts, while only the selected k experts are executed. As a result, globally balanced routing can still assign substantial probability mass to unselected experts. We refer to this probability mass outside the selected top-k set as *routing leakage*, reflecting a mismatch between the router distribution and the sparse computation it drives. To address it, we introduce a routing objective that penalizes leakage outside the selected experts ($\mathcal{L}_{\text{leak}}$), encourages probability sharing for low-margin tokens and sharper assignments for high-margin tokens within the selected set ($\mathcal{L}_{\text{adaptive}}$), and maintains balanced expert utilization across the batch ($\mathcal{L}_{\text{util}}$).

We call the resulting method Fisher-Adaptive MoE (FA-MoE) and evaluate it on three LVLM backbones—StableLM-1.6B, Qwen-1.8B, and Phi-2 (2.7B)—across eight multimodal benchmarks. Across three seeds, FA-MoE consistently improves MME and SEED-Bench over the standard auxiliary-routing baseline (V-MoE) across all three backbones. On MME, the mean improvements are 53.7, 49.4, and 62.9 points on StableLM, Qwen, and Phi-2, respectively. Comparisons with recent routing and dense-to-MoE upcycling methods further show that these gains are not specific to the standard auxiliary-routing baseline. Routing analyses further show lower routing leakage and fewer subsequent changes in the selected top-2 expert set, while different experts receive different proportions of text and image tokens.

In summary, our main contributions are:

1. **Representation-Aware Router Initialization.** We use pretrained dense representations to initialize the MoE router with Fisher-discriminative directions instead of discarding this information during upcycling.
2. **Token-Level Routing Objective.** We introduce a routing objective that reduces probability leakage, adapts uncertainty within the selected experts, and maintains balanced expert utilization.
3. **Evaluation and Routing Analysis.** Across three LVLM backbones and eight multimodal benchmarks, FA-MoE achieves the highest score outright in 21 of 24 settings. Controlled ablations support the initialization and objective choices, while routing analyses show lower leakage and fewer changes in selected expert sets during training.

2 METHODOLOGY

FA-MoE modifies dense-to-MoE upcycling at two points: how the router is initialized before sparse training, and how its routing distribution is regularized during training. We first define the sparse routing setup, then introduce Fisher-initialized routing (Section 2.2) and the token-level routing objective (Section 2.3).

2.1 PRELIMINARIES

Consider an MoE layer with E expert FFNs $\{f_i^{\mathrm{FFN}}\}_{i=1}^{E}$. Given a token representation $\mathbf{x}_t$, a router with weight rows $\mathbf{w}_1, \ldots, \mathbf{w}_E$ produces logits $\mathbf{r}_t \in \mathbb{R}^E$. The MoE-LLaVA baseline uses a linear router, $r_{t,i} = \mathbf{w}_i^\top \mathbf{x}_t$. The routing probabilities are

$$p_{t,i} = \frac{\exp(r_{t,i})}{\sum_{j=1}^{E} \exp(r_{t,j})}. \tag{1}$$

Let

$$\mathcal{T}_k(t) = \mathrm{TopK}(\mathbf{p}_t, k) \tag{2}$$

denote the k experts with the largest routing probabilities.

After selecting the top-k experts, we renormalize their probabilities as

$$\hat{p}_{t,i} = \frac{p_{t,i}}{\sum_{j \in \mathcal{T}_k(t)} p_{t,j}}, \qquad i \in \mathcal{T}_k(t), \tag{3}$$

and compute

$$\mathrm{MoE}(\mathbf{x}_t) = \sum_{i \in \mathcal{T}_k(t)} \hat{p}_{t,i}\, f_i^{\mathrm{FFN}}(\mathbf{x}_t). \tag{4}$$

The V-MoE baseline is trained with

$$\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{task}} + \beta \mathcal{L}_{\mathrm{aux}}, \tag{5}$$

where $\mathcal{L}_{\mathrm{task}}$ is the autoregressive task loss and

$$\mathcal{L}_{\mathrm{aux}} = E \cdot \sum_{i=1}^{E} F_i G_i \tag{6}$$

is the standard auxiliary load-balancing loss. Here, E is the number of experts, F_i is the fraction of tokens whose top-1 routed expert is i, and $G_i = \frac{1}{T}\sum_{t=1}^{T} p_{t,i}$ is the mean routing probability assigned to expert i over the T tokens in the batch. We use $\beta = 0.01$.

2.2 FISHER-INITIALIZED ROUTING

In dense-to-MoE upcycling, each expert inherits the pretrained dense FFN weights, whereas the router is newly introduced and is typically initialized randomly. We therefore use the structure already present in the dense model's hidden representations to initialize the router.

For each transformer layer converted into an MoE layer, we collect token representations from the corresponding dense model layer and partition them into E groups with K-means. The assignments serve only as pseudo-labels; the K-means centroids are not used as router weights. We then seek directions that separate the group means while accounting for variation within each group. With S_B and S_W denoting the between- and within-group scatter matrices (Appendix H), these directions maximize Fisher's criterion

$$J(\mathbf{v}) = \frac{\mathbf{v}^\top S_B \mathbf{v}}{\mathbf{v}^\top S_W \mathbf{v}}, \tag{7}$$

and satisfy the generalized eigenproblem $S_B\mathbf{v} = \eta S_W \mathbf{v}$, as in Fisher's linear discriminant analysis (LDA) (Fisher, 1936).

With E pseudo-classes, Fisher LDA provides at most $E - 1$ non-zero discriminative directions. We obtain the remaining router direction by orthogonal completion. Because each Fisher eigenvector defines an axis up to sign, we orient each direction toward the centered cluster mean with which it has the largest absolute alignment before using the resulting directions as router rows. Algorithm 1 and Appendix H provide the complete router initialization procedure.

Cosine routing. Fisher's criterion is invariant to the scale of $\mathbf{v}$, so the resulting solutions are defined by their direction rather than their norm. We therefore use cosine similarity for routing, making each score depend on the angular alignment between the token representation and a Fisher direction:

$$r_{t,i} = s \left\langle \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|_2}, \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2} \right\rangle, \tag{8}$$

where $s = 10$ is a fixed logit scale; its motivation and implementation details are provided in Appendix H.6. Cosine routing has also been used in prior MoE work to reduce sensitivity to representation magnitude (Chi et al., 2022; Nguyen et al., 2025).

As illustrated in Figure 1(c), Fisher initialization produces expert assignments that better follow the local separations already present in the dense representation space.

2.3 Token-Level Routing Objective

Fisher initialization provides a structured starting point, but routing probabilities continue to evolve during sparse training. Standard load balancing controls average expert usage across a batch, but does not directly regulate each token's routing distribution. We therefore regularize routing at two levels: leakage and adaptive entropy control the token-level distribution outside and within the selected top-k set, respectively, while utilization maintains balanced expert usage across the batch.

Routing leakage. For each token, only the experts in $\mathcal{T}_k(t)$ participate in the sparse forward pass. We define the probability mass outside this selected set as

$$\mathcal{L}_{\text{leak}} = \frac{1}{T} \sum_{t=1}^{T} \left(1 - \sum_{j \in \mathcal{T}_k(t)} p_{t,j} \right). \tag{9}$$

Because the selected probabilities are renormalized before combining expert outputs (Eq. 3), leakage does not change their relative mixture weights once the top-k set is fixed. Instead, it measures how much probability remains assigned to experts that are not executed. We penalize this mass because it reflects routing uncertainty outside the selected expert set. Empirically, tokens with larger leakage are more likely to change their selected top-k experts at later training checkpoints (Appendix E). For the main setting $E = 4$ and $k = 2$, uniform routing corresponds to leakage 0.5.

Margin-aware adaptive entropy. Leakage measures probability assigned outside the selected expert set. We separately regulate how probability is shared *within* that set using the renormalized probabilities from Eq. 3. Their entropy is

$$H_t = - \sum_{i \in \mathcal{T}_k(t)} \hat{p}_{t,i} \log \hat{p}_{t,i}, \qquad H_t \in [0, \log k]. \tag{10}$$

Let $p_{t,(1)}$ and $p_{t,(2)}$ denote the largest and second-largest routing probabilities. We use their margin

$$m_t = p_{t,(1)} - p_{t,(2)} \tag{11}$$

as a measure of routing confidence and convert it into an ambiguity weight

$$\alpha_t = (1 - m_t)^{\gamma}. \tag{12}$$

Here, γ controls how quickly the ambiguity weight decreases as the routing margin increases. We use a fixed value in all experiments; its setting is provided in Appendix H. During optimization, α_t is detached from the computational graph and acts only as a gating weight.

The adaptive entropy loss is

$$\mathcal{L}^{(t)}_{\text{adaptive}} = \alpha_t (\log k - H_t) + (1 - \alpha_t) H_t. \tag{13}$$

The loss changes its entropy preference according to routing confidence. Low-margin tokens are encouraged to share probability across the selected experts, whereas high-margin tokens are encouraged to form a more concentrated assignment. For a low-margin token, forcing a sharp choice between nearly tied experts can commit the router arbitrarily and concentrate the routing weight on one expert despite the routing uncertainty; allowing probability sharing lets both selected experts receive training signal. Thus, the objective does not uniformly minimize entropy for every token.

Batch-level utilization. Leakage and adaptive entropy act at the token level. To maintain balanced expert usage across the batch, we define the batch-averaged routing probability

$$u_i = \frac{1}{T} \sum_{t=1}^{T} p_{t,i} \tag{14}$$

and penalize its deviation from uniform utilization:

$$\mathcal{L}_{\text{util}} = \sum_{i=1}^{E} \left(u_i - \frac{1}{E} \right)^2 . \tag{15}$$

Complete objective. The routing objective combines the token-level and batch-level terms:

$$\mathcal{L}_{\text{router}} = \mathcal{L}_{\text{leak}} + \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{\text{adaptive}}^{(t)} + \mathcal{L}_{\text{util}}. \tag{16}$$

The three routing terms are weighted equally within $\mathcal{L}_{\text{router}}$. FA-MoE is trained with

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{router}}. \tag{17}$$

We use a fixed routing-loss coefficient in all main experiments; its value and sensitivity are reported in Appendix D.3.

3 Related Work

Sparse MoE routing and load balancing. Token-choice sparse MoE models route each token to a subset of experts, typically with an auxiliary load-balancing loss to prevent expert collapse (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022); this regulates aggregate expert usage but not the shape of each token's routing distribution. Expert-choice routing instead lets each expert select a fixed number of tokens for controlled capacity (Zhou et al., 2022). Other work targets training stability: ST-MoE adds a router z-loss (Zoph et al., 2022), StableMoE distills and freezes a learned routing strategy (Dai et al., 2022), and Guo et al. (2025) show that auxiliary balancing can cause expert overlap and introduce orthogonality and variance losses to encourage specialization. FA-MoE instead replaces the standard auxiliary loss with a routing objective that combines token-level leakage and adaptive-entropy terms with batch-level expert-utilization regularization.

Routing geometry and expert specialization. X-MoE scores routing on a low-dimensional hypersphere to reduce representation collapse (Chi et al., 2022); FA-MoE also uses cosine-based scoring but initializes router directions from the dense model's representation geometry. ACMoE reweights features to separate latent clusters (Nielsen et al., 2025), while ERMoE routes using similarity to learned orthonormal expert eigenbases (Cheng et al., 2026). DeepSeekMoE promotes specialization through fine-grained expert segmentation and shared experts (Dai et al., 2024), while Krishnamurthy et al. (2023) introduce explicit specialization regularization. OpenMoE observes that token-to-expert assignments are established early in pretraining and remain largely unchanged (Xue et al., 2024). FA-MoE differs by using dense-model representation geometry to initialize the router before sparse training, together with a token-level routing objective during training.

Using the dense model to initialize or guide the router. Sparse upcycling copies dense FFN weights into experts (Komatsuzaki et al., 2023), while Drop-Upcycling partially reinitializes them to encourage differentiation (Nakamura et al., 2025). In LVLMs, MoE-LLaVA upcycles language-model FFNs (Lin et al., 2026), while CuMo co-upcycles MoE blocks in the vision encoder and connector (Li et al., 2024b). Recent work also uses dense-model information to initialize or supervise routing. Router Upcycling initializes routers from attention heads in the dense checkpoint (Ran et al., 2025), and Teacher-Guided Routing supervises a student router using routing distributions from a frozen dense teacher throughout training (Kada et al., 2026). Most closely related, Cluster-Aware Upcycling clusters dense activations, initializes the router from cluster centroids, constructs cluster-specific expert weights via truncated SVD, and adds an expert-ensemble self-distillation loss (Chu et al., 2026). FA-MoE instead uses K-means assignments only as pseudo-labels to compute Fisher-discriminative router directions, rather than using the centroids directly, and uses the dense model only for initialization.

4 RESULTS

4.1 EXPERIMENTAL SETUP

We build on the MoE-LLaVA dense-to-MoE framework (Lin et al., 2026) and evaluate FA-MoE across three backbone language models: StableLM-1.6B (Bellagente et al., 2024), Qwen-1.8B (Bai et al., 2023), and Phi-2 (2.7B) (Javaheripi et al., 2023). Our main experiments use $E = 4$ experts per MoE layer. V-MoE and FA-MoE use top-2 routing following the MoE-LLaVA setup, while StableMoE uses top-1 routing according to its original design. We additionally evaluate $E = 8$ with top-2 routing in Appendix D.2 to examine FA-MoE with a larger expert pool.

Within each comparison, we keep the backbone checkpoint, multimodal training data, optimization schedule, and evaluation pipeline fixed while retaining each method's routing or upcycling mechanism. V-MoE denotes the standard linear top-k router trained with the auxiliary load-balancing objective. StableMoE (Dai et al., 2022) uses auxiliary load balancing during its first stage and freezes the distilled router during the second stage. FA-MoE replaces the auxiliary load-balancing loss with the routing objective introduced in Section 2.3.

FA-MoE results in Table 1 are reported as the mean over three seeds; the corresponding per-seed results and sample standard deviations are provided in Appendix C. We additionally compare against X-MoE, Drop-Upcycling, and Cluster-Aware Upcycling (CA-MoE) across all three backbones using the same dense-to-MoE training and evaluation pipeline while retaining each method's specific mechanism.

We evaluate all models with zero-shot inference on eight multimodal benchmarks: GQA (Hudson & Manning, 2019), ScienceQA-IMG (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al., 2023b), MME (Fu et al., 2023), SEED-Bench (Li et al., 2024a), MMStar (Chen et al., 2024), and MMBench (Liu et al., 2024). Appendix B provides the hardware and optimization settings, sparse-layer configuration, dense-to-MoE initialization protocol, and evaluation setup used in these experiments.

4.2 BENCHMARK EVALUATION

Performance across backbones. Table 1 shows that FA-MoE ranks first in 21 of the 24 backbone–benchmark settings and ties for first in one additional setting. The most consistent gains over V-MoE occur on MME, where FA-MoE improves by 53.7, 49.4, and 62.9 points on StableLM, Qwen, and Phi-2, respectively. SEED-Bench also improves across all three backbones, by 3.3, 0.8, and 0.9 points. The per-seed results and standard deviations for these comparisons are reported in Appendix C.

Beyond these two benchmarks, FA-MoE matches or improves upon the V-MoE mean on all reported backbone–metric combinations except Phi-2 MMBench. The differences are small on several metrics—for example, Qwen GQA is unchanged at the displayed precision—while larger gains appear on metrics such as ScienceQA-IMG and MMStar for some backbones. On Phi-2 MMBench, V-MoE scores 76.2 compared with 75.0 for FA-MoE. Overall, FA-MoE achieves the highest score in 21 of the 24 backbone–benchmark comparisons.

4.3 COMPARISON WITH RECENT ROUTING AND UPCYCLING METHODS

We next compare FA-MoE with recent methods that modify routing or dense-to-MoE upcycling. The comparison includes X-MoE (Chi et al., 2022), Drop-Upcycling (Nakamura et al., 2025), and Cluster-Aware Upcycling (Chu et al., 2026). Each method is adapted to the same dense-to-MoE training and evaluation pipeline across all three backbones while retaining its method-specific mechanism. Implementation details for these adaptations are provided in the appendix.

Table 1 provides a comparison with methods that address different aspects of sparse routing and upcycling. On StableLM, relative to Cluster-Aware Upcycling, the FA-MoE three-seed mean is higher by 3.2 points on GQA, 2.6 on ScienceQA-IMG, 1.6 on TextVQA, 0.7 on POPE, and 42.2 on MME. Cluster-Aware Upcycling is especially relevant because it also uses dense-model activations to initialize sparse routing. However, it initializes the router from cluster centroids and modifies the experts using cluster-specific truncated SVD, whereas FA-MoE retains the copied experts and uses

Table 1: Comparison of routing and dense-to-MoE upcycling methods across three LVLM backbones. SQA^I denotes ScienceQA-IMG and VQA^T denotes TextVQA. Bold indicates the highest reported score and underline indicates the second-highest reported score within each backbone and benchmark.

Backbone	Method	GQA	SQA^I	VQA^T	POPE	MME	SEED	MMStar	MMB
StableLM	V-MoE	60.5	59.6	49.8	86.1	1544.2	61.9	30.1	70.5
	StableMoE	58.2	57.4	44.2	85.4	1511.2	61.0	29.7	67.9
	X-MoE	58.0	56.4	48.6	85.3	1512.8	61.2	28.6	68.5
	Drop-Upcycling	59.1	58.3	47.9	86.1	1528.1	**69.1**	29.9	70.0
	CA-MoE	58.2	57.4	49.1	85.7	1555.7	63.1	28.7	70.5
	FA-MoE	**61.4**	**60.0**	**50.7**	**86.4**	**1597.9**	65.2	**31.2**	**72.0**
Qwen	V-MoE	**61.6**	61.5	48.8	86.9	1513.0	63.2	32.4	70.7
	StableMoE	59.1	60.5	47.4	86.6	1509.9	61.6	30.0	68.8
	X-MoE	59.3	58.2	47.9	85.8	1542.8	62.7	31.3	69.9
	Drop-Upcycling	57.2	58.7	44.8	85.5	1503.1	62.1	30.5	68.2
	CA-MoE	60.4	59.9	47.1	86.2	1527.3	63.4	32.2	69.5
	FA-MoE	**61.6**	**62.8**	**48.9**	**87.2**	**1562.4**	**64.0**	**33.7**	**70.9**
Phi-2	V-MoE	61.6	70.6	52.7	86.3	1598.5	64.6	34.4	**76.2**
	StableMoE	60.1	67.8	49.2	84.6	1550.5	63.0	31.3	68.4
	X-MoE	58.3	66.6	50.2	86.2	1589.8	62.4	35.1	69.4
	Drop-Upcycling	59.7	65.3	50.8	85.2	1599.2	61.8	33.6	70.4
	CA-MoE	60.8	70.1	51.2	86.4	1568.1	63.2	35.0	72.6
	FA-MoE	**61.7**	**71.4**	**53.0**	**87.3**	**1661.4**	**65.5**	**37.2**	75.0

Table 2: **Component analysis on StableLM-1.6B.** All configurations are matched single-run experiments with the same training and evaluation setup. (a) Router initialization under matched routing objectives. (b) Routing-objective ablation with Fisher initialization fixed. **Full** denotes margin-aware adaptive entropy + routing leakage + batch-level utilization. Bold indicates the highest score and underline indicates the second-highest score within each panel and benchmark.

(a) Router initialization

Init.	Objective	GQA	SQA^I	VQA^T	POPE	MME
Random	Auxiliary	59.40	59.80	49.10	85.90	1555.10
Fisher	Auxiliary	60.12	**60.37**	49.42	85.74	1564.83
Random	Full	59.23	59.39	48.89	86.21	1547.57
Centroid	Full	60.27	58.15	50.28	86.25	1566.28
Fisher	Full	**61.47**	59.92	**50.86**	**86.93**	**1594.28**

(b) Routing objective with Fisher initialization

Objective	GQA	SQA^I	VQA^T	POPE	MME
Auxiliary	60.12	**60.37**	49.42	85.74	1564.83
Adaptive entropy	60.21	60.18	50.13	86.02	1557.95
Routing leakage	60.28	60.24	49.98	86.27	1548.67
Adaptive + Leakage	60.02	60.01	50.58	86.21	1585.32
Adaptive + Utilization	59.69	59.73	50.80	86.84	1544.36
Full	**61.47**	59.92	**50.86**	**86.93**	**1594.28**

Fisher-discriminative directions for router initialization. We therefore examine the centroid-versus-Fisher distinction directly under a matched router setup in the next section.

4.4 Component Analysis

Table 2 separates the effects of router initialization and the routing objective using matched experiments on StableLM-1.6B. Every matched control in Table 2 uses this same cosine router. Thus, the matched comparisons separate the effects of router initialization and the routing objective from the use of cosine routing itself. Random+Auxiliary is therefore a matched cosine-router control and is distinct from the V-MoE baseline in Table 1, which retains its native linear router. For the centroid control, the K-means cluster centroids are used directly as router rows.

Router initialization. Panel (a) compares Fisher initialization with random and centroid initialization while holding the routing objective fixed within each comparison. Under the auxiliary objective, Fisher improves GQA, ScienceQA-IMG, TextVQA, and MME relative to random initialization, while POPE changes only slightly. With the full routing objective, Fisher initialization exceeds both

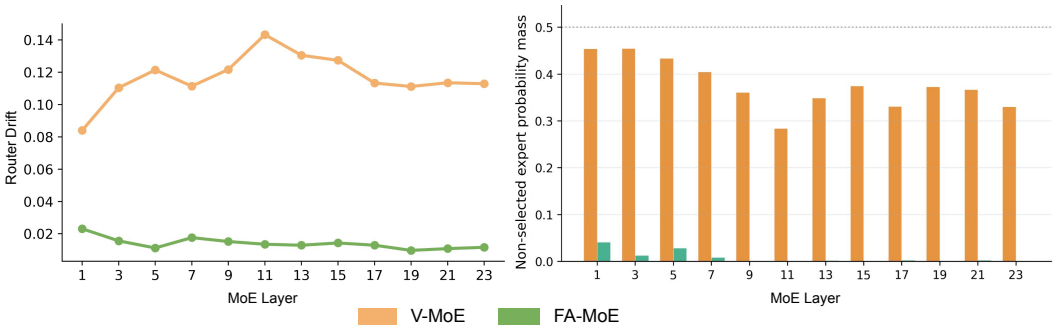

Figure 2: **Router drift and routing leakage across MoE layers.** Left: normalized router drift from initialization to the final checkpoint. Right: mean probability mass assigned outside the selected top-2 experts at the final checkpoint. V-MoE and FA-MoE differ in router parameterization, initialization, and routing objective, so the drift comparison characterizes the complete routing systems.

Random+Full and Centroid+Full on all five reported metrics. In particular, relative to Centroid+Full, Fisher+Full improves GQA by 1.20, ScienceQA-IMG by 1.77, TextVQA by 0.58, POPE by 0.68, and MME by 28.00. This comparison is important because both initializations use the same clustered dense representations, but only Fisher initialization uses discriminative directions rather than cluster centroids as router weights.

Routing objective and interaction with initialization. Panel (b) fixes Fisher initialization and varies the routing objective. The individual routing terms do not consistently improve performance over the auxiliary objective when used alone. Partial combinations also exhibit mixed behavior: Adaptive + Leakage substantially improves MME over either term alone, whereas Adaptive + Utilization improves TextVQA and POPE but reduces GQA and MME. The full objective achieves the highest GQA, TextVQA, POPE, and MME scores, reaching 61.47, 50.86, 86.93, and 1594.28, respectively. ScienceQA-IMG is the exception, where the auxiliary objective remains highest at 60.37.

The effect of the routing objective also depends on initialization. Replacing the auxiliary objective with the full objective under random initialization does not improve GQA or MME, whereas the same change under Fisher initialization increases GQA from 60.12 to 61.47 and MME from 1564.83 to 1594.28. Relative to Adaptive + Leakage, adding batch-level utilization in the full objective further increases MME from 1585.32 to 1594.28 and improves GQA, TextVQA, and POPE, while ScienceQA-IMG decreases slightly. These results suggest that the routing terms interact rather than contributing independent gains on every benchmark, and that their strongest overall performance is obtained when combined with Fisher initialization. A direct comparison between ordinary and margin-aware adaptive entropy is provided in Appendix D.1.

4.5 Routing Behavior

We next examine the routing behavior of FA-MoE across transformer layers. We first analyze routing leakage and router drift, then examine whether the resulting routing remains distributed across experts rather than collapsing to a small subset.

Layer-wise routing leakage and router drift. Figure 2 compares routing leakage and normalized router drift across the evaluated MoE layers. Router drift is the Frobenius-norm change in the router weights from initialization to the final checkpoint, normalized by the initialization norm. FA-MoE assigns substantially less probability mass outside the selected top-2 experts and shows lower drift across layers; the mean drift over the 12 MoE layers decreases from 0.1168 for V-MoE to 0.0140 for FA-MoE (Appendix G).

Leakage and subsequent routing changes. We further examine whether routing leakage is associated with later changes in the selected expert set. For each pair of consecutive saved checkpoints, which are 100 training steps apart, we measure a token's leakage at the current checkpoint and

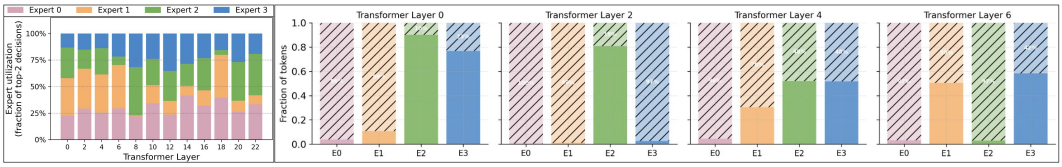

Figure 3: **Expert utilization and modality-dependent routing in FA-MoE.** Left: fraction of top-2 routing decisions assigned to each expert across transformer layers on ScienceQA. All four experts continue to receive routed tokens, while their utilization varies across layers. Right: text and image token composition among routed tokens for selected layers. Different experts receive different proportions of text and image tokens, indicating modality-dependent allocation patterns.

record whether its exact top-2 expert set changes at the next checkpoint. We report Spearman's rank correlation coefficient ρ between current per-token leakage and this binary top-2-set-change indicator, pooled across all layers and consecutive checkpoint pairs. Higher leakage is positively associated with a later top-2-set change for both FA-MoE ($\rho = 0.185$) and V-MoE ($\rho = 0.227$). We define the consecutive-checkpoint flip rate as the fraction of tokens whose exact top-2 expert set changes between consecutive checkpoints. FA-MoE has a lower mean flip rate, 0.074 compared with 0.086 for V-MoE. These observations show an association between leakage and subsequent routing changes; we do not claim that reducing leakage is the causal source of improved downstream performance.

Expert utilization and modality-dependent routing. More concentrated routing could still lead to expert collapse if most tokens were assigned to only a small subset of experts. Figure 3 shows that all four experts continue to receive tokens across the evaluated layers, while their relative utilization varies across the network. Thus, the routing is neither collapsed to a small subset of experts nor fixed uniformly across layers.

The routed token composition also varies across experts. For the selected layers, different experts receive different proportions of text and image tokens, indicating modality-dependent allocation patterns without any explicit modality assignment. These differences indicate modality-dependent routing preferences, but are not interpreted as evidence of semantic expert specialization.

Together, these analyses show that FA-MoE maintains low routing leakage across layers while continuing to use all experts. The lower router drift and the association between leakage and subsequent top-2-set changes provide additional evidence about how its routing evolves during sparse training. Additional analyses of expert count, router-loss strength, Fisher-subspace stability, orthogonal-completion sensitivity, and supporting visualizations are provided in the appendix.

5 Conclusions, Limitations, and Future Work

We introduced FA-MoE which initializes MoE router directions from the Fisher-discriminative geometry of a pretrained dense model's hidden representations and trains the router with an objective that controls probability mass outside the selected experts, adapts entropy according to routing confidence, and maintains balanced expert utilization. Across three LVLM backbones and eight multimodal benchmarks, FA-MoE ranks first in 21 of 24 backbone–benchmark settings. Controlled ablations further show that Fisher initialization improves over random and centroid initialization under the same routing objective. Our routing analysis also shows lower leakage and fewer subsequent changes in the selected top-2 expert set, suggesting that the representation structure already present in a pretrained dense model provides a useful basis for learning sparse routing during MoE upcycling.

Our experiments focus on compact LVLM backbones below 3B parameters. Future work can investigate whether the same initialization and routing principles extend to substantially larger LVLMs, larger expert pools, different sparsity levels, and other sparse-routing architectures. The layer-wise analyses also indicate that the structure captured by Fisher initialization is clearer in earlier layers than at greater depth, motivating depth-aware router initialization as another direction for future work.

AI USE STATEMENT

We used generative AI tools (an LLM-based assistant) solely for writing and editing assistance on the manuscript text, including proofreading, fixing internal cross-references and citation-label errors, and improving clarity and phrasing of existing content. We did not use generative AI tools to generate research ideas, design experiments, write or modify the experimental code and models, run experiments, or analyze results; all methodology, experiments, and result analysis in this paper are the authors' own work. All AI-assisted edits to the text were reviewed by the authors, who take full responsibility for the final content of this work, including all text, claims, and reported results.

ETHICS STATEMENT

This work builds on publicly released pretrained language and vision-language models and publicly available benchmark datasets (VQAv2, GQA, TextVQA, ScienceQA, MMBench, POPE, MME, MM-Vet, and SEED-Bench), all of which are standard, widely used academic evaluation resources; we did not collect new data or conduct any human-subjects research. We are not aware of potential harmful applications, fairness, privacy, or security concerns specific to this work beyond those already associated with the underlying pretrained models and public benchmarks, and we do not foresee conflicts of interest or sponsorship issues related to this submission.

REPRODUCIBILITY STATEMENT

We describe the complete FA-MoE method, including Fisher-discriminative router initialization and the token-level routing objective, in Section 2, with the full derivation and additional Fisher geometry analyses in Appendix H. The complete experimental protocol, hardware, optimization settings, and evaluation benchmarks are given in Appendix B, and full three-seed results corresponding to the means reported in the main paper are provided in Appendix C. Anonymous, downloadable source code and scripts sufficient to reproduce all reported experiments are provided at the link in Appendix B.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. Stable LM 2 1.6B technical report. *arXiv preprint arXiv:2402.17834*, 2024.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087, 2024. doi: 10.52202/079017-0850.

Anzhe Cheng, Shukai Duan, Shixuan Li, Chenzhong Yin, Mingxi Cheng, Heng Ping, Tamoghna Chattopadhyay, Sophia I Thomopoulos, Shahin Nazarian, Paul Thompson, and Paul Bogdan. Ermoe: Eigen-reparameterized mixture-of-experts for stable routing and interpretable specialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12997–13006, 2026.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34600–34613, 2022.

Sanghyeok Chu, Pyunghwan Ahn, Gwangmo Song, Seung Hwan Kim, Honglak Lee, and Bohyung Han. Enhancing mixture-of-experts specialization via cluster-aware upcycling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11283–11292, 2026.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7085–7095, 2022.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.70.

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.

Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Xinye Cao, Sicong Leng, Qimei Cui, et al. Advancing expert specialization for better moe. *arXiv preprint arXiv:2505.22323*, 2025.

Ethan He, Abhinav Khattar, Ryan Prenger, Vijay Korthikanti, Zijie Yan, Tong Liu, Shiqing Fan, Ashwath Aithal, Mohammad Shoeybi, and Bryan Catanzaro. Upcycling large language models into mixture of experts. *arXiv preprint arXiv:2410.07524*, 2024.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

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

Masahiro Kada, Ryota Yoshihashi, Satoshi Ikehata, Rei Kawakami, and Ikuro Sato. Teacher-guided routing for sparse vision mixture-of-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6198–6208, 2026.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *International Conference on Learning Representations*, 2023.

Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. Improving expert specialization in mixture of experts. *arXiv preprint arXiv:2302.14703*, 2023.

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

Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. In *International Conference on Learning Representations*, 2023a.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13299–13308, 2024a.

Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. CuMo: Scaling multimodal LLM with co-upcycled mixture-of-experts. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-4170.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.20.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. MoE-LLaVA: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, 28:4408–4419, 2026. doi: 10.1109/TMM.2026.3654458.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, volume 15064 of *Lecture Notes in Computer Science*, pp. 216–233. Springer, 2024. doi: 10.1007/978-3-031-72658-3_13.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2507–2521, 2022.

Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. Drop-Upcycling: Training sparse mixture of experts with partial re-initialization. In *International Conference on Learning Representations*, 2025.

Huy Nguyen, Pedram Akbarian, Trang Pham, Trang Nguyen, Shujian Zhang, and Nhat Ho. Statistical advantages of perturbing cosine router in mixture of experts. In *International Conference on Learning Representations*, 2025.

Stefan Nielsen, Rachel Teo, Laziz Abdullaev, and Tan Nguyen. Tight clusters make specialized experts. In *International Conference on Learning Representations*, 2025.

Junfeng Ran, Guangxiang Zhao, Yuhan Wu, Dawei Zhu, Longyun Wu, Yikai Zhao, Tong Yang, Lin Sun, Xiangzheng Zhang, and Sujian Li. Router upcycling: Leveraging mixture-of-routers in mixture-of-experts upcycling. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pp. 4241–4248, 2025. doi: 10.3233/FAIA251318.

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.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. NormFace: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1041–1049, 2017. doi: 10.1145/3123266.3123359.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. OpenMoE: An early effort on open mixture-of-experts language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55625–55655. PMLR, 2024.

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, pp. 7103–7114, 2022.

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. *arXiv preprint arXiv:2202.08906*, 2022.

A Appendix Overview

This appendix provides additional experimental results, implementation details, and routing analyses that complement the main paper. Appendix B gives the complete experimental protocol and motivates the compact LVLM setting used in this study. Appendix C reports three-seed variability for FA-MoE, and Appendix D provides additional ablations on the entropy objective, expert count, routing-loss coefficient, and Fisher orthogonal completion. Appendix E analyzes routing concentration, leakage, and subsequent changes in the selected expert set. Appendices F and G provide additional expert-allocation and router-stability analyses. Appendix H gives the complete Fisher-initialization procedure, explains why Fisher directions differ from centroid initialization, and analyzes the stability of the resulting geometry. Appendix I summarizes simple bounds on the router-loss terms. Appendix J extends the router-geometry visualization to multiple layers and all three backbones.

B Detailed Experimental Setup

Reproducibility. The code and scripts needed to reproduce the experiments are provided at `anonymous.4open.science/r/FA-MoE-5DC8`.

Hardware and optimization. All experiments were conducted on a server equipped with 8 NVIDIA A100-SXM4-40GB GPUs. Training used DeepSpeed v0.9.5 with ZeRO Stage 2, BF16 precision, and gradient checkpointing. StableLM and Qwen experiments used 3 GPUs with a per-device batch size of 2 and gradient accumulation over 12 steps. Phi-2 experiments used 4 GPUs with a per-device batch size of 1 and gradient accumulation over 12 steps. All models were trained for one epoch on the LLaVA-FT instruction-tuning mixture. We used AdamW with learning rate 2×10^{-5}, cosine learning-rate scheduling, warmup ratio 0.03, weight decay 0.0, maximum sequence length $2{,}048$, and group-by-modality-length batching. Table 3 summarizes the compute and optimization configuration used for the main experiments.

Model architecture. We build on the MoE-LLaVA dense-to-MoE conversion pipeline. For the principal experiments, sparse MoE layers replace the FFN layers in alternating transformer blocks and contain $E = 4$ experts. V-MoE and FA-MoE use top-2 routing. StableMoE retains its native top-1 routing configuration. The backbone, multimodal data, optimization schedule, and evaluation pipeline are otherwise held fixed within each comparison.

Backbones and dense-to-MoE initialization. We evaluate three LVLM backbones: StableLM-1.6B, Qwen-1.8B, and Phi-2 (2.7B). For all three backbones, the dense Stage-II LVLM checkpoint is used as the starting point for Stage-III MoE training. Expert parameters are initialized by copying the corresponding dense FFN weights. FA-MoE additionally initializes the router using Fisher-discriminative directions obtained from dense-model hidden representations, with K-means assignments used only as pseudo-labels.

Why compact LVLM backbones. We focus on LVLM backbones below 3B parameters because improving compact multimodal models is an increasingly important setting for resource-constrained training and deployment. This regime also makes it possible to evaluate FA-MoE across multiple backbones, seeds, routing baselines, and controlled ablations under a practical compute budget. StableLM and Qwen training uses three A100 GPUs, while the largest training configuration, Phi-2 (2.7B), uses four A100 GPUs. The routing hyperparameters used in the main FA-MoE experiments are summarized in Table 4.

Table 3: Compute and optimization configuration used for the main experiments.

Setting	**Value**
GPU hardware	NVIDIA A100-SXM4-40GB
Available GPUs	8
GPU memory	40GB per GPU
StableLM/Qwen training	3 GPUs, batch size 2 per GPU
Phi-2 training	4 GPUs, batch size 1 per GPU
Gradient accumulation	12 steps
Effective batch size, StableLM/Qwen	72
Effective batch size, Phi-2	48
Distributed training	DeepSpeed v0.9.5, ZeRO Stage 2
Precision	BF16
Gradient checkpointing	Enabled
Optimizer	AdamW
Learning rate	2×10^{-5}
Learning-rate schedule	Cosine
Warmup ratio	0.03
Weight decay	0.0
Training duration	1 epoch
Maximum sequence length	2,048 tokens
Fisher initialization overhead	approximately 12.5–30 minutes

Table 4: Routing hyperparameters used in the main FA-MoE experiments.

Setting	Value
Number of experts E	4
Selected experts k	2
Cosine logit scale s	10
Adaptive-entropy exponent γ	2
Routing-loss coefficient λ	0.01
V-MoE auxiliary coefficient β	0.01

Auxiliary-loss protocol. The standard auxiliary load-balancing loss is active for the V-MoE baseline with coefficient 0.01. StableMoE uses the auxiliary objective during its first training stage; after the distilled router is frozen, its second stage optimizes only the task objective. The standard auxiliary load-balancing loss is disabled only for FA-MoE, where it is replaced by

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{router}}, \qquad \lambda = 0.01 \tag{18}$$

unless otherwise stated.

Router parameterization across comparisons. The broad method comparison preserves the native routing mechanism of each baseline. In particular, the competing methods are not forced to use FA-MoE's cosine router or fixed logit scale. FA-MoE uses cosine routing with fixed scale $s = 10$. In contrast, the matched initialization–objective ablations in the main paper use the same cosine router and fixed scale for every cell, so that only initialization and routing objective vary.

Sparse forward pass. All FA-MoE experiments renormalize the routing probabilities over the selected top-k expert set before combining expert outputs. For token t,

$$\hat{p}_{t,i} = \frac{p_{t,i}}{\sum_{j \in \mathcal{T}_k(t)} p_{t,j}}, \qquad i \in \mathcal{T}_k(t), \tag{19}$$

and

$$\text{MoE}(\mathbf{x}_t) = \sum_{i \in \mathcal{T}_k(t)} \hat{p}_{t,i} f_i^{\text{FFN}}(\mathbf{x}_t). \tag{20}$$

Thus, probability leakage is a property of the full router distribution; it does not change which experts are executed once the top-k set has been selected, nor the relative mixture weights after selected-set renormalization. We therefore use leakage as a routing-distribution diagnostic rather than as a direct measure of task performance.

Table 5: Full three-seed results for FA-MoE across the three LVLM backbones. Values are reported as mean ± sample standard deviation over three complete runs.

Backbone	GQA	SQA^I	VQA^T	POPE	MME	SEED	MMStar	MMB
StableLM	61.4 ± 0.1	60.0 ± 0.1	50.7 ± 0.2	86.4 ± 0.2	1597.9 ± 8.7	65.2 ± 0.2	31.2 ± 0.1	72.0 ± 0.3
Qwen	61.6 ± 0.2	62.8 ± 0.3	48.9 ± 0.3	87.2 ± 0.2	1562.4 ± 9.1	64.0 ± 0.2	33.7 ± 0.4	70.9 ± 0.2
Phi-2	61.7 ± 0.2	71.4 ± 0.3	53.0 ± 0.3	87.3 ± 0.1	1661.4 ± 16.3	65.5 ± 0.2	37.2 ± 0.4	75.0 ± 0.2

Table 6: Ordinary versus margin-aware adaptive entropy under the matched FA-MoE setup. The adaptive row corresponds to the Fisher+Full configuration used in the main component analysis.

Entropy	GQA	SQA^I	VQA^T	POPE	MME
Ordinary	57.20	57.30	47.90	85.90	1558.10
Adaptive	**61.47**	**59.92**	**50.86**	**86.93**	**1594.28**

Recent routing and upcycling baselines. X-MoE, Drop-Upcycling, and Cluster-Aware Upcycling are integrated into the same dense-to-MoE LVLM training and evaluation pipeline while retaining their method-specific routing or upcycling mechanisms. We do not add FA-MoE's Fisher initialization or routing loss to these methods. The main paper reports the resulting comparison under the same downstream evaluation scripts.

Routing diagnostics. In addition to downstream benchmark scores, we measure probability leakage, top-k routing entropy, top-1 confidence, expert utilization, router drift, router-direction alignment, and changes in the selected top-k set across training checkpoints.

Evaluation benchmarks. We evaluate models using zero-shot inference on eight multimodal benchmarks: GQA, ScienceQA-IMG, TextVQA, POPE, MME, SEED-Bench, MMStar, and MMBench. GQA measures compositional visual question answering; ScienceQA-IMG evaluates multimodal science reasoning; TextVQA evaluates reading and reasoning over text appearing in images; POPE evaluates object hallucination; MME covers a broad collection of perception and cognition tasks; SEED-Bench measures generative multimodal comprehension; MMStar emphasizes vision-dependent multimodal reasoning; and MMBench evaluates a broad set of perceptual and reasoning capabilities.

C Three-Seed Variability of FA-MoE

The main paper reports three-seed means for FA-MoE to keep the comparison table readable. Table 5 reports the corresponding sample standard deviations. These runs use the same training and evaluation protocol across seeds.

Table 5 shows consistent performance across three independent runs for all three backbones. The clearest repeated gains over V-MoE occur on MME and SEED-Bench. On MME, FA-MoE improves the mean by 53.7, 49.4, and 62.9 points for StableLM, Qwen, and Phi-2, respectively. On SEED-Bench, the corresponding gains are $+3.3$, $+0.8$, and $+0.9$.

D Additional Ablations and Scope Experiments

D.1 Ordinary versus Adaptive Entropy

The main paper uses a margin-aware adaptive entropy objective. To test whether ordinary entropy minimization is sufficient, we keep Fisher initialization, leakage regularization, batch-level utilization regularization, cosine routing, fixed logit scale, training budget, and evaluation fixed, and change only the entropy formulation.

As shown in Table 6, adaptive entropy improves over ordinary entropy on all five evaluated metrics under the same Fisher initialization and routing configuration, with gains of 4.27 GQA, 2.62 ScienceQA-IMG, 2.96 TextVQA, 1.03 POPE, and 36.18 MME.

Table 7: Effect of the number of experts on StableLM with top-2 routing. The $E = 4$ rows are three-seed means; the $E = 8$ rows are single-run results.

E	Method	GQA	SQA^{I}	VQA^{T}	MME
4	V-MoE	60.5	59.6	49.8	1544.2
	FA-MoE	**61.4**	**60.0**	**50.7**	**1597.9**
8	V-MoE	59.3	59.8	49.4	1557.5
	FA-MoE	**61.5**	**60.4**	**51.3**	**1603.6**

Table 8: Sensitivity to the routing-loss coefficient λ across the three backbones.

Backbone	λ	GQA	SQA^{I}	MME	SEED
StableLM	0.1	62.1	59.4	1564.6	61.1
	0.01	61.4	59.9	1594.2	65.2
Qwen	0.1	62.5	61.9	1501.0	62.5
	0.01	61.8	62.7	1561.5	63.8
Phi-2	0.1	61.1	68.2	1683.2	64.1
	0.01	61.7	71.1	1661.7	65.3

D.2 Effect of the Number of Experts

We additionally evaluate a larger $E = 8$ configuration on StableLM while retaining top-2 routing shown in Table 7. Thus, two experts are activated per token in both the $E = 4$ and $E = 8$ configurations.

As shown in Table 7, FA-MoE retains its advantage over V-MoE when the number of experts increases from $E = 4$ to $E = 8$ while keeping top-2 routing fixed.

D.3 Effect of the Router-Loss Coefficient

We also study the router-loss coefficient λ in a separate coefficient-sensitivity sweep. Table 8 reports the results for $\lambda = 0.01$ and $\lambda = 0.1$ across the three backbones.

Across these runs, $\lambda = 0.01$ gives the more consistent profile across metrics and backbones, although $\lambda = 0.1$ is higher on individual metrics, such as StableLM GQA, Qwen GQA, and Phi-2 MME. We therefore use $\lambda = 0.01$ as a light routing regularizer rather than allowing the routing objective to dominate the task objective.

D.4 Sensitivity to Orthogonal Completion

For $E = 4$ pseudo-classes, Fisher LDA provides at most $E - 1 = 3$ non-zero discriminative directions. The fourth router row is therefore obtained through orthogonal completion. To test whether the resulting initial partition is dominated by this random completion, we sample 50 orthogonal completions while holding the remaining Fisher directions and all other factors fixed.

As shown in Table 9, the initial routing partition is stable across the 50 sampled orthogonal completions, with 91.98% pairwise exact top-2 agreement and only 1.26 percentage points of variation in Expert-4 inclusion.

E Routing Dynamics and Leakage Analysis

E.1 Routing Concentration Across Training

Figure 4 compares the routing distributions of V-MoE and FA-MoE. For V-MoE, 0.39 of the routing probability lies outside the selected top-2 set at initialization and 0.38 at the end of training, compared with 0.5 under uniform routing. Its full-distribution entropy also remains close to the uniform-routing

Table 9: Sensitivity of the initial routing partition to the fourth router row across 50 orthogonal completions.

Measure	Value
Pairwise exact top-2-set agreement	91.98%
Std. of Expert-4 inclusion rate	1.26 percentage points

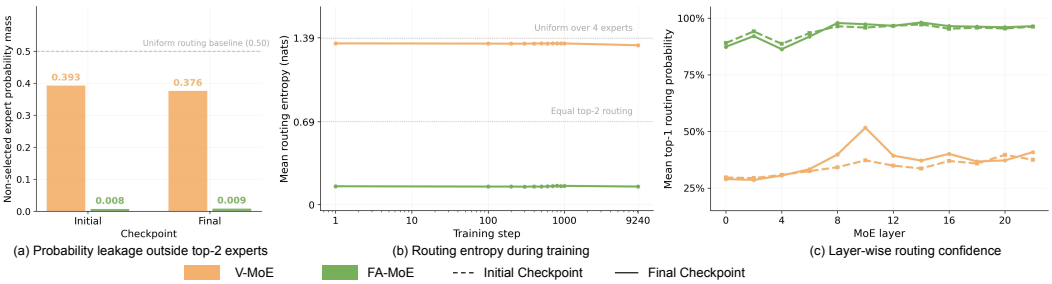

Figure 4: **Token-level routing behavior of V-MoE and FA-MoE on StableLM-1.6B.** (a) Probability mass outside the selected top-2 experts (routing leakage) at the initial and final checkpoints; uniform routing gives leakage 0.5. (b) Entropy of the full routing distribution during training; uniform routing over four experts gives $\log 4 \approx 1.39$. (c) Top-1 routing probability across layers at the initial (dashed) and final (solid) checkpoints. Because the two methods use different router parameterizations, these plots characterize the complete routing systems; the controlled component ablations in the main paper isolate initialization and objective choices.

value, while its top-1 probability stays below 0.5. FA-MoE instead keeps leakage below 0.01 at both checkpoints. Because FA-MoE already has low leakage at initialization, the initial difference reflects Fisher initialization together with the cosine router, whereas the final difference additionally reflects the routing objective.

Interpretation of leakage. Leakage is directly optimized by the FA-MoE objective, so lower leakage alone should not be interpreted as evidence that it causes better downstream performance. We use it as a diagnostic of how probability mass is distributed outside the selected computation path. To provide an additional behavioral check, we relate leakage to a quantity that the objective does not directly optimize: whether the selected expert set changes later in training.

E.2 Leakage and Subsequent Expert-Set Changes

For a token evaluated at checkpoint c, we measure its leakage at c and record whether its exact top-2 expert set changes at checkpoint $c + 100$. Current leakage is positively associated with a later routing-set change for both V-MoE and FA-MoE, and the future flip rate increases monotonically across leakage quartiles for both methods.

Table 10 shows that higher leakage is positively associated with a later change in the selected top-2 expert set for both V-MoE and FA-MoE. FA-MoE also has a lower mean consecutive-checkpoint flip rate (0.074 versus 0.086). This analysis characterizes routing behavior; we do not claim that reducing leakage is the causal source of downstream performance gains.

F Modality-Dependent Expert Allocation

We provide layer-wise text/image routing statistics for StableLM, Qwen, and Phi-2 in Figures 5, 6, and 7. All experts receive routed tokens, while the proportions of text and image tokens differ across experts and layers. These plots demonstrate differentiated allocation of multimodal tokens without hard-coded modality assignments. We do not interpret these modality proportions alone as evidence of semantic expert specialization.

Table 10: Leakage and subsequent routing-set changes. ρ measures the association between current leakage and a later exact top-2-set change. The flip rate is the mean fraction of tokens whose selected top-2 set changes between consecutive checkpoints.

Method	ρ	**Consecutive-checkpoint flip rate ↓**
V-MoE	0.227	0.086
FA-MoE	0.185	0.074

Table 11: Router stability across MoE layers on StableLM-1.6B. Lower drift and higher alignment indicate stronger preservation of the initial router parameters.

MoE Layer	**Router Drift ↓**		**Direction Alignment ↑**	
	V-MoE	**FA-MoE**	**V-MoE**	**FA-MoE**
0	0.0840	0.0231	0.9965	0.9997
2	0.1104	0.0156	0.9939	0.9999
4	0.1214	0.0112	0.9926	0.9999
6	0.1114	0.0176	0.9938	0.9998
8	0.1217	0.0152	0.9927	0.9999
10	0.1433	0.0135	0.9899	0.9999
12	0.1305	0.0129	0.9916	0.9999
14	0.1274	0.0143	0.9919	0.9999
16	0.1133	0.0129	0.9936	0.9999
18	0.1112	0.0097	0.9938	1.0000
20	0.1135	0.0108	0.9936	0.9999
22	0.1129	0.0116	0.9937	0.9999
Global	**0.1168**	**0.0140**	**0.9931**	**0.9999**

G Router Stability Analysis

We measure how much each router changes between initialization and the final checkpoint. Normalized router drift is defined as

$$\text{Drift} = \frac{\left\|W_r^{\text{final}} - W_r^{\text{init}}\right\|_F}{\left\|W_r^{\text{init}}\right\|_F}. \tag{21}$$

We also measure the average direction alignment between corresponding initial and final router rows:

$$\text{Align} = \frac{1}{E}\sum_{i=1}^{E} \cos\left(\mathbf{w}_i^{\text{init}}, \mathbf{w}_i^{\text{final}}\right). \tag{22}$$

Table 11 shows that FA-MoE has lower router drift at every reported layer, with global drift decreasing from 0.1168 for V-MoE to 0.0140 for FA-MoE. Direction alignment shows the same trend. Because V-MoE and FA-MoE use different complete routing systems, these values characterize end-to-end routing behavior rather than initialization alone.

H Supplementary Details for Fisher-Initialized Routing

This section gives the complete implementation procedure for the Fisher-initialized router. The procedure uses the dense model's hidden-state geometry to initialize each future MoE router before sparse training. K-means is used only to produce activation-space pseudo-labels; its centroids are not used as the FA-MoE router weights.

H.1 Activation Collection

For each transformer layer that will be converted into an MoE layer, we collect token-level hidden representations from the corresponding dense Stage-II LVLM before sparsification. Let $\mathcal{D}$ denote a

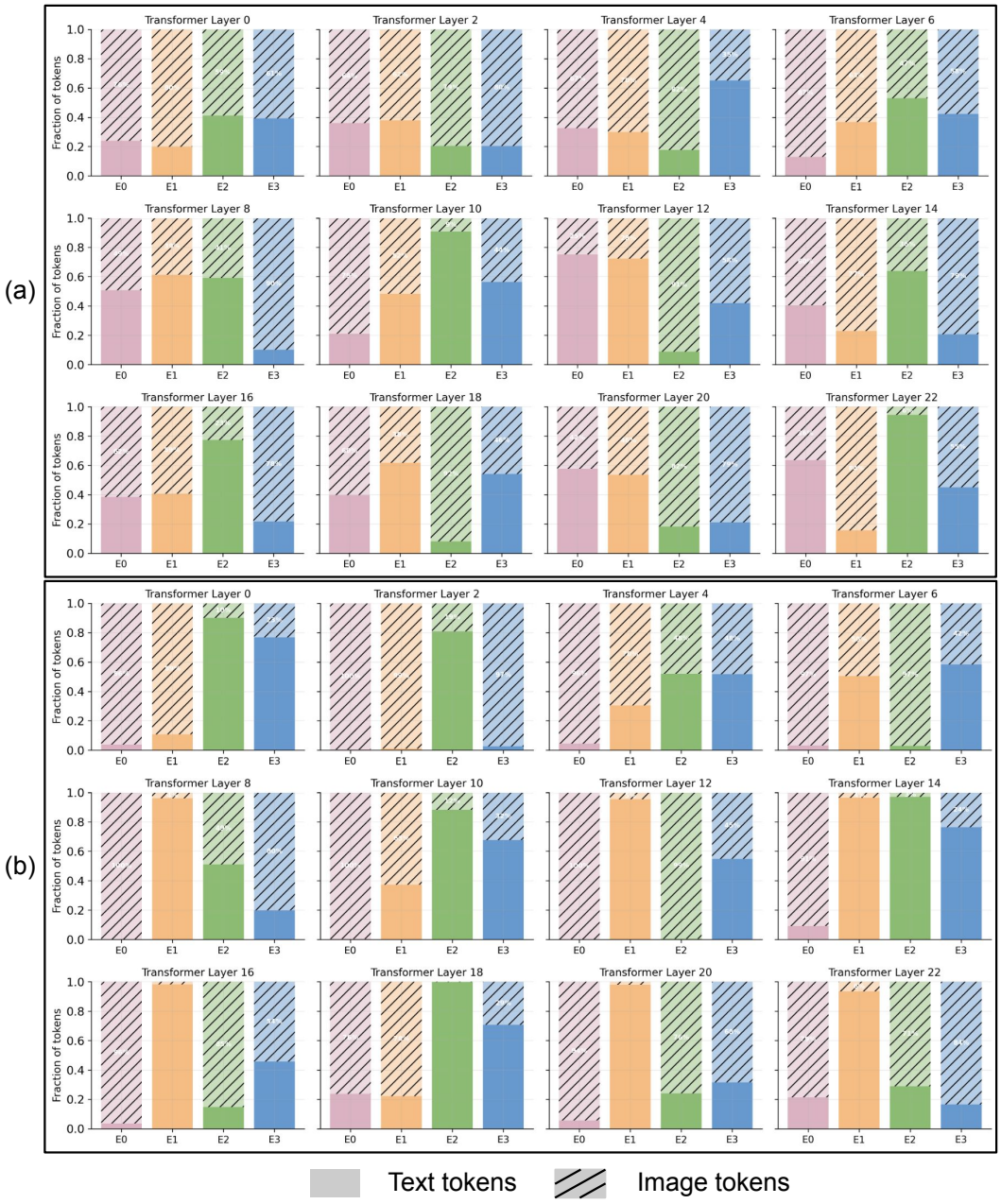

Figure 5: **StableLM modality-dependent expert allocation.** Fraction of text and image tokens routed to each expert across MoE layers.

subset of the instruction-tuning data, and let $\mathbf{x}_t^{(\ell)} \in \mathbb{R}^d$ denote the token representation at layer ℓ. We collect

$$\mathcal{X}^{(\ell)} = \left\{ \mathbf{x}_t^{(\ell)} \right\}_{t=1}^{N_\ell}. \tag{23}$$

For the router initialization used in the main experiments, we randomly sample approximately $5{,}000$ token representations per future MoE layer. K-means assigns pseudo-labels to these representations, after which Fisher LDA is used to estimate discriminative directions.

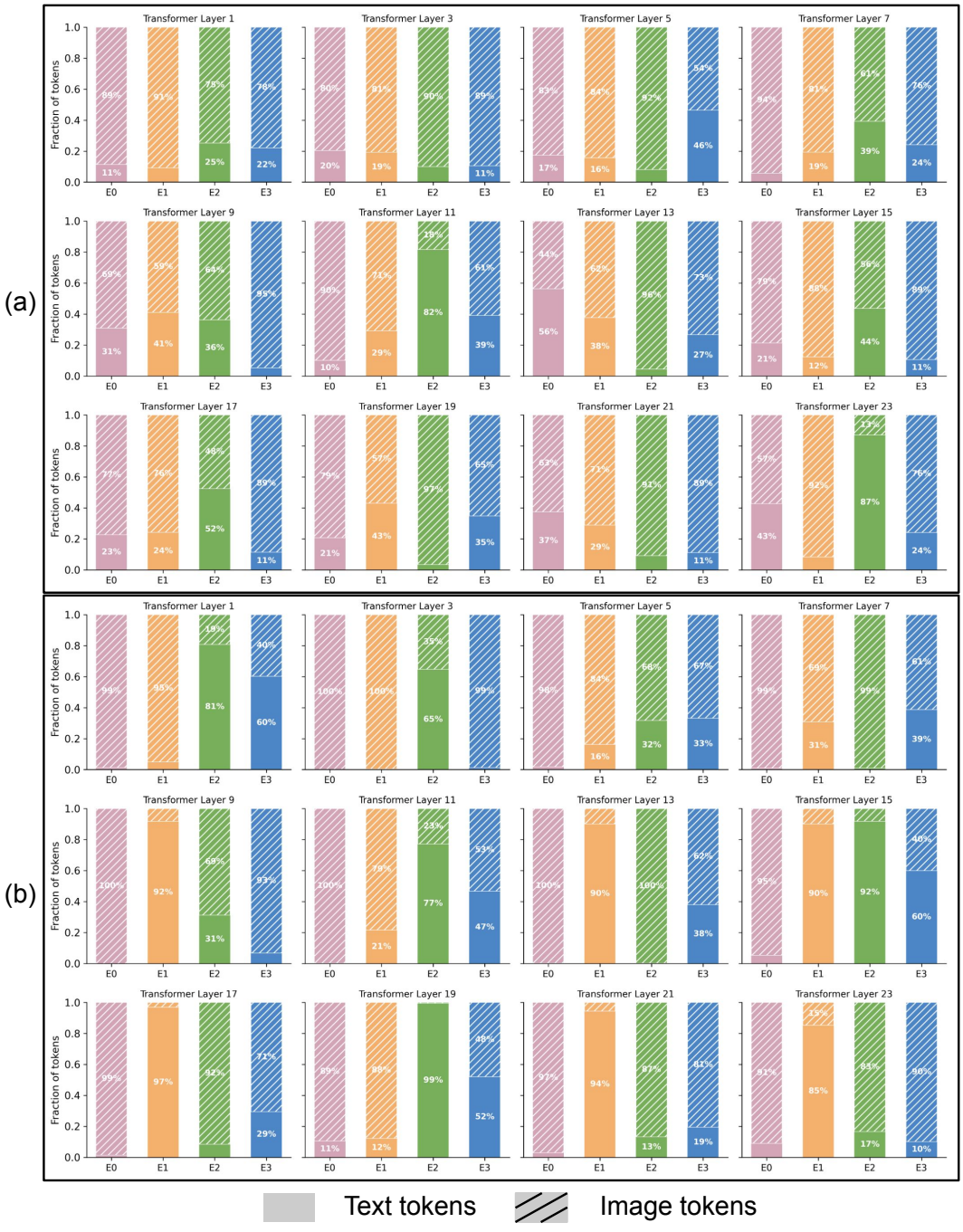

Figure 6: **Qwen modality-dependent expert allocation.** Fraction of text and image tokens routed to each expert across MoE layers.

H.2 K-Means Pseudo-Labels

For layer ℓ, K-means with E clusters gives

$$y_t^{(\ell)} = \mathrm{KMeans}\left(\mathbf{x}_t^{(\ell)}\right), \qquad y_t^{(\ell)} \in \{1, \ldots, E\}. \tag{24}$$

The assignments serve only as pseudo-labels. Let $\mathcal{X}_c^{(\ell)}$ denote the representations assigned to cluster c, with $n_c^{(\ell)} = |\mathcal{X}_c^{(\ell)}|$. The cluster and global means are

$$\boldsymbol{\mu}_c^{(\ell)} = \frac{1}{n_c^{(\ell)}} \sum_{\mathbf{x} \in \mathcal{X}_c^{(\ell)}} \mathbf{x}, \tag{25}$$

and

$$\bar{\boldsymbol{\mu}}^{(\ell)} = \frac{1}{N_\ell} \sum_{t=1}^{N_\ell} \mathbf{x}_t^{(\ell)}. \tag{26}$$

H.3 Fisher Discriminative Directions

The within-cluster scatter matrix is

$$S_W^{(\ell)} = \sum_{c=1}^{E} \sum_{\mathbf{x} \in \mathcal{X}_c^{(\ell)}} \left(\mathbf{x} - \boldsymbol{\mu}_c^{(\ell)}\right) \left(\mathbf{x} - \boldsymbol{\mu}_c^{(\ell)}\right)^\top, \tag{27}$$

and the between-cluster scatter matrix is

$$S_B^{(\ell)} = \sum_{c=1}^{E} n_c^{(\ell)} \left(\boldsymbol{\mu}_c^{(\ell)} - \bar{\boldsymbol{\mu}}^{(\ell)}\right) \left(\boldsymbol{\mu}_c^{(\ell)} - \bar{\boldsymbol{\mu}}^{(\ell)}\right)^\top. \tag{28}$$

Fisher directions solve

$$S_B v = \eta S_W v. \tag{29}$$

For numerical stability, we use

$$\left(S_W + \epsilon I\right)^{-1} S_B v = \eta v, \tag{30}$$

with $\epsilon = 10^{-4}$.

For E pseudo-classes, Fisher LDA provides at most $E - 1$ non-zero discriminative directions. We therefore retain

$$\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{E-1}^{(\ell)} \tag{31}$$

corresponding to the largest generalized eigenvalues. The final router direction is obtained by sampling a random vector and orthogonalizing it against these $E - 1$ Fisher directions:

$$\mathbf{v}_E^{(\ell)} \perp \operatorname{span}\left(\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{E-1}^{(\ell)}\right). \tag{32}$$

H.4 Why Fisher Directions Instead of Cluster Centroids

A centroid and a Fisher direction capture different properties of the activation space. A centroid describes where a group of hidden representations is located, whereas Fisher LDA identifies directions that emphasize separation between groups relative to variation within them.

This distinction is relevant to FA-MoE because its cosine router assigns scores according to the angular alignment between a token representation and a router row. Using a normalized K-means centroid as a router row therefore uses the direction from the origin to the center of a cluster. This direction summarizes the cluster location, but it is not chosen to distinguish that cluster from the others. Fisher directions, in contrast, are explicitly obtained from between-group and within-group scatter and therefore emphasize discriminative directions in the representation space.

The controlled initialization comparison in the main paper supports this distinction empirically: with the cosine router, routing objective, training budget, and evaluation fixed, Fisher initialization performs better than centroid initialization on all five evaluated metrics.

H.5 Direction Orientation and Normalization

Fisher eigenvectors define axes, so both $\mathbf{v}$ and $-\mathbf{v}$ are valid solutions. For each Fisher direction $\mathbf{v}_q^{(\ell)}$, we identify the most aligned centered cluster mean:

$$c^* = \arg\max_c \left| \left\langle \mathbf{v}_q^{(\ell)}, \boldsymbol{\mu}_c^{(\ell)} - \bar{\boldsymbol{\mu}}^{(\ell)} \right\rangle \right|. \tag{33}$$

We orient the direction toward this cluster:

$$\mathbf{v}_q^{(\ell)} \leftarrow \begin{cases} -\mathbf{v}_q^{(\ell)}, & \text{if } \left\langle \mathbf{v}_q^{(\ell)}, \boldsymbol{\mu}_{c^*}^{(\ell)} - \bar{\boldsymbol{\mu}}^{(\ell)} \right\rangle < 0, \\ \mathbf{v}_q^{(\ell)}, & \text{otherwise.} \end{cases} \tag{34}$$

All router directions are then normalized:

$$\tilde{\mathbf{v}}_q^{(\ell)} = \frac{\mathbf{v}_q^{(\ell)}}{\left\| \mathbf{v}_q^{(\ell)} \right\|_2}, \qquad q = 1, \ldots, E. \tag{35}$$

After orientation and normalization, the E directions are stacked as the rows of the router weight matrix. This construction should not be interpreted as fitting a multiclass LDA classifier. Fisher LDA is used only to obtain a discriminative subspace from the activation-space pseudo-labels; the resulting directions are converted into router rows by orientation and orthogonal completion. Likewise, there is no fixed one-cluster-per-expert assignment. At inference and training time, every token is scored against all router rows, and the top-k experts are selected from these scores.

H.6 Cosine Routing and Logit Scale

Cosine similarity is bounded in $[-1, 1]$. Without an additional logit scale, this range limits how concentrated the resulting softmax distribution can become. For example, with $E = 4$ experts, the largest possible top-1 probability is

$$\frac{e}{e + 3e^{-1}} = \frac{e^2}{e^2 + 3} \approx 0.71, \tag{36}$$

and the largest possible probability mass on the top-2 experts is

$$\frac{2e}{2e + 2e^{-1}} = \frac{e^2}{e^2 + 1} \approx 0.88. \tag{37}$$

Thus, unscaled cosine logits would impose a minimum leakage of approximately 0.12 in our $E = 4$, top-2 setting.

Normalized-softmax classifiers similarly use a scale parameter to avoid an overly restricted softmax range (Wang et al., 2017), and cosine MoE routers use temperature or score scaling for the same purpose (Chi et al., 2022; Li et al., 2023a). We use a fixed logit scale $s = 10$ in all FA-MoE experiments and in every cell of the matched initialization–objective controls.

The normalized Fisher directions initialize the router rows as

$$W_r^{(\ell)} = s \begin{bmatrix} \tilde{\mathbf{v}}_1^{(\ell)} \\ \vdots \\ \tilde{\mathbf{v}}_E^{(\ell)} \end{bmatrix}. \tag{38}$$

During training, router rows are projected back to norm s:

$$\mathbf{w}_i^{(\ell)} \leftarrow s \frac{\mathbf{w}_i^{(\ell)}}{\|\mathbf{w}_i^{(\ell)}\|_2}. \tag{39}$$

The routing score itself uses the normalized row through Eq. 8; consequently, its magnitude is controlled by the explicit logit scale s, while the routing direction is determined by angular similarity.

Table 12: Stability and held-out discriminativeness of the rank-3 Fisher subspace at layer 0 across five repetitions. Values are mean ± standard deviation.

Measure	Value
Fisher subspace overlap	0.7708 ± 0.0049
Mean principal-angle cosine	0.8754 ± 0.0028
Held-out J, Fisher subspace	7.343 ± 5.323
Held-out J, equal-rank random subspace	0.367 ± 0.184
Fisher/random J ratio	$18.94 \pm 11.45\times$

H.7 Stability of the Fisher Geometry

We test whether the estimated Fisher geometry is reproducible across independently sampled activations and remains informative on held-out examples. For this analysis, two independently sampled, example-disjoint $5{,}000$-example subsets are drawn from the same dense checkpoint and instruction mixture. Example provenance is preserved so that fitting and held-out evaluation sets do not share examples.

For $E = 4$, we analyze only the true rank-$(E-1) = 3$ Fisher subspace and exclude the orthogonal-completion row. Across five repetitions at layer 0, we compare the Fisher subspaces estimated from the two disjoint activation samples and evaluate the source-Fisher projection on held-out examples.

To evaluate whether the learned Fisher subspace remains discriminative on held-out activations, let $Q \in \mathbb{R}^{d\times(E-1)}$ denote the orthonormal Fisher basis estimated from the source split. The K-means model fitted on the source split is then used to assign a pseudo-label $y_i \in \{1, \dots, E\}$ to each held-out representation $\mathbf{x}_i$. We project each held-out representation into the Fisher subspace,

$$\mathbf{z}_i = Q^\top \mathbf{x}_i. \tag{40}$$

Let $\bar{\mathbf{z}}_c$ denote the mean of the projected held-out examples assigned to pseudo-class c, let n_c be the number of such examples, and let $\bar{\mathbf{z}} = \frac{1}{N}\sum_i \mathbf{z}_i$. We define the held-out subspace discriminativeness score as

$$J(Q) = \frac{\sum_{c=1}^{E} n_c \left\|\bar{\mathbf{z}}_c - \bar{\mathbf{z}}\right\|_2^2}{\sum_{c=1}^{E}\sum_{i:\, y_i=c} \left\|\mathbf{z}_i - \bar{\mathbf{z}}_c\right\|_2^2}. \tag{41}$$

This is the between-class to within-class scatter ratio evaluated in the rank-$(E-1)$ projected subspace on the held-out split. For the random-subspace baseline, the same score is computed after replacing Q with an equal-rank random orthonormal basis.

As shown in Table 12, Fisher subspaces estimated from independent activation samples have substantial overlap and remain strongly discriminative on held-out examples. The held-out subspace score is 7.343 ± 5.323 for the Fisher subspace, compared with 0.367 ± 0.184 for an equal-rank random subspace.

H.8 Relation to a Pretrained MoE Router

FA-MoE addresses dense-to-MoE conversion, where the starting checkpoint has no router. If a compatible pretrained MoE checkpoint is available, its router could instead provide initialization or supervision. Such router transfer is complementary to the setting studied here: FA-MoE derives a routing prior directly from hidden representations of the dense model and does not require a separate MoE teacher, pre-existing routing labels, or an additional router-training stage.

H.9 Initialization Overhead

Fisher initialization is performed once before sparse MoE training. In our implementation, constructing all router weights requires approximately 12.5 minutes for the StableLM-1.6B model and approximately 30 minutes for the Phi-2 (2.7B) model. This includes activation collection, K-means pseudo-label construction, Fisher-direction computation, and router-weight construction. The initialized weights are saved and reused during sparse training, so this cost is not incurred during each training epoch.

Algorithm 1 Fisher-Initialized Directional Routing

Require: Dense model before sparsification, instruction-tuning subset $\mathcal{D}$, number of experts E, future MoE layers $\mathcal{L}_{\mathrm{MoE}}$

Ensure: Initialized router weights $\{W_r^{(\ell)}\}_{\ell \in \mathcal{L}_{\mathrm{MoE}}}$

for $\ell \in \mathcal{L}_{\mathrm{MoE}}$ **do**
  Collect token representations $\mathcal{X}^{(\ell)}$ from the dense model on $\mathcal{D}$.
  Run K-means with E clusters to obtain activation-space pseudo-labels.
  Compute cluster means $\{\boldsymbol{\mu}_c^{(\ell)}\}_{c=1}^{E}$ and global mean $\bar{\boldsymbol{\mu}}^{(\ell)}$.
  Form within-cluster scatter $S_W^{(\ell)}$ and between-cluster scatter $S_B^{(\ell)}$.
  Solve the Fisher eigenproblem and retain the top $E-1$ discriminative directions.
  Obtain the remaining direction by orthogonal completion.
  Orient each direction toward its most aligned centered cluster mean.
  Normalize all directions.
  Initialize router rows with the normalized directions and fixed scale s.
end for
Train the sparse MoE model with cosine routing and the proposed routing objective.

Algorithm 1 summarizes the complete Fisher-initialized routing procedure.

I Boundary Conditions of the FA-MoE Router Objective

We summarize the numerical bounds and limiting behavior of the three router-loss components.

Leakage loss. For each token t,

$$\ell_{\mathrm{leak}}^{(t)} = 1 - \sum_{j \in \mathcal{T}_k(t)} p_{t,j}. \tag{42}$$

Since $\mathcal{T}_k(t)$ contains the k largest entries of $\mathbf{p}_t \in \Delta_E$,

$$\sum_{j \in \mathcal{T}_k(t)} p_{t,j} \geq \frac{k}{E}. \tag{43}$$

Therefore,

$$0 \leq \ell_{\mathrm{leak}}^{(t)} \leq \frac{E-k}{E}, \qquad 0 \leq \mathcal{L}_{\mathrm{leak}} \leq \frac{E-k}{E}. \tag{44}$$

For $E = 4$ and $k = 2$, uniform routing attains the maximum leakage of 0.5.

Adaptive entropy loss. For token t,

$$\mathcal{L}_{\mathrm{adaptive}}^{(t)} = \alpha_t(\log k - H_t) + (1-\alpha_t)H_t, \tag{45}$$

where $\alpha_t \in [0,1]$ and $H_t \in [0, \log k]$. Both H_t and $\log k - H_t$ lie in $[0, \log k]$, so

$$0 \leq \mathcal{L}_{\mathrm{adaptive}}^{(t)} \leq \log k. \tag{46}$$

For top-2 routing,

$$0 \leq \mathcal{L}_{\mathrm{adaptive}}^{(t)} \leq \log 2. \tag{47}$$

Batch-level utilization. Using the notation of the main paper,

$$\mathcal{L}_{\mathrm{util}} = \sum_{i=1}^{E} \left(u_i - \frac{1}{E} \right)^2, \qquad \sum_{i=1}^{E} u_i = 1. \tag{48}$$

The minimum is attained at uniform batch-averaged utilization, $u_i = 1/E$. The maximum occurs when all batch-averaged probability mass is assigned to one expert:

$$0 \leq \mathcal{L}_{\mathrm{util}} \leq 1 - \frac{1}{E}. \tag{49}$$

For $E = 4$, the upper bound is 0.75.

Summary. The three terms therefore satisfy

$$0 \le \mathcal{L}_{\text{leak}} \le \frac{E-k}{E}, \qquad 0 \le \mathcal{L}_{\text{adaptive}}^{(t)} \le \log k, \qquad 0 \le \mathcal{L}_{\text{util}} \le 1 - \frac{1}{E}. \tag{50}$$

J Layer-wise Routing Structure Across Backbones

Figure 1 in the main text shows the routing structure for one layer. Figures 8, 9, and 10 show the same comparison at layers 0, 2, 10, and 22 for Qwen-1.8B, Phi-2, and StableLM-1.6B.

Across all three backbones, Fisher-initialized routing follows the visible text–image structure more closely in the early layers. At later layers, the difference between Fisher-initialized and random routing becomes less clear. This suggests that the visible text–image structure captured by Fisher initialization is clearer in earlier layers and becomes less distinct at greater depth.

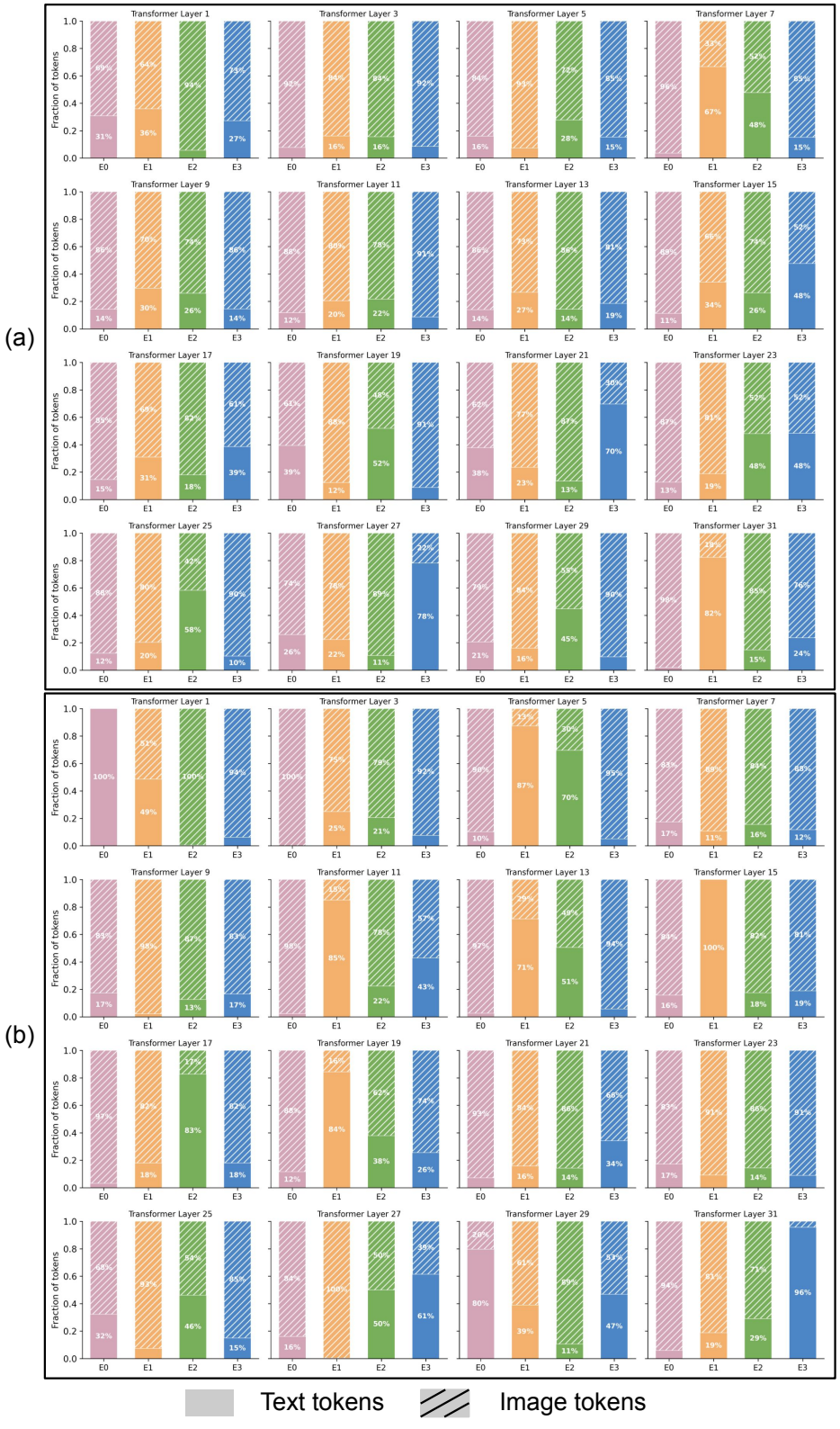

Figure 7: **Phi-2 modality-dependent expert allocation.** Fraction of text and image tokens routed to each expert across MoE layers.

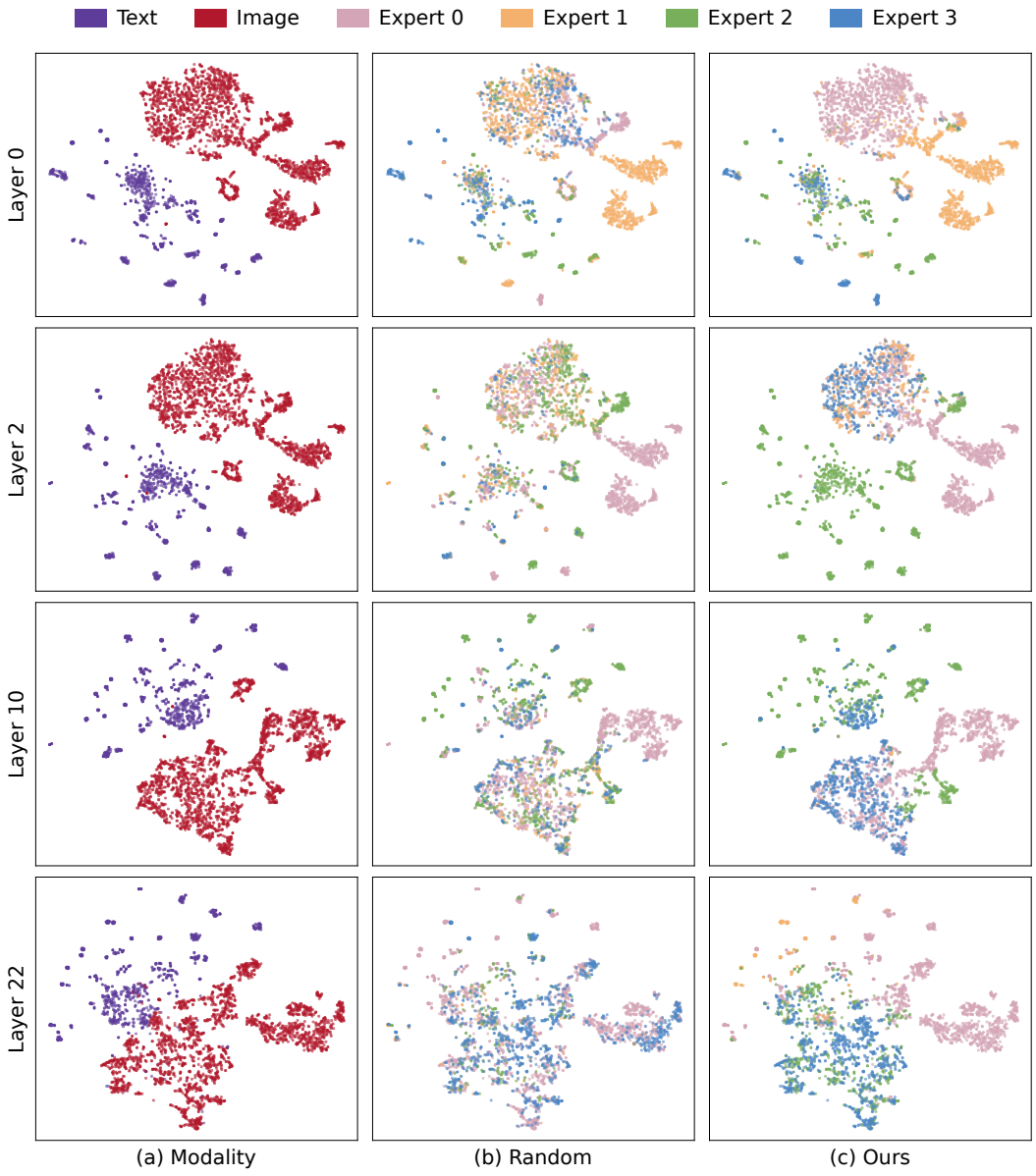

Figure 8: Layer-wise comparison of modality structure (a), random routing (b), and Fisher-initialized routing (c) for **Qwen-1.8B** at layers 0, 2, 10, and 22. In the early layers, Fisher-initialized routing follows the visible text–image structure more closely than random routing. This difference becomes less clear in the later layers.

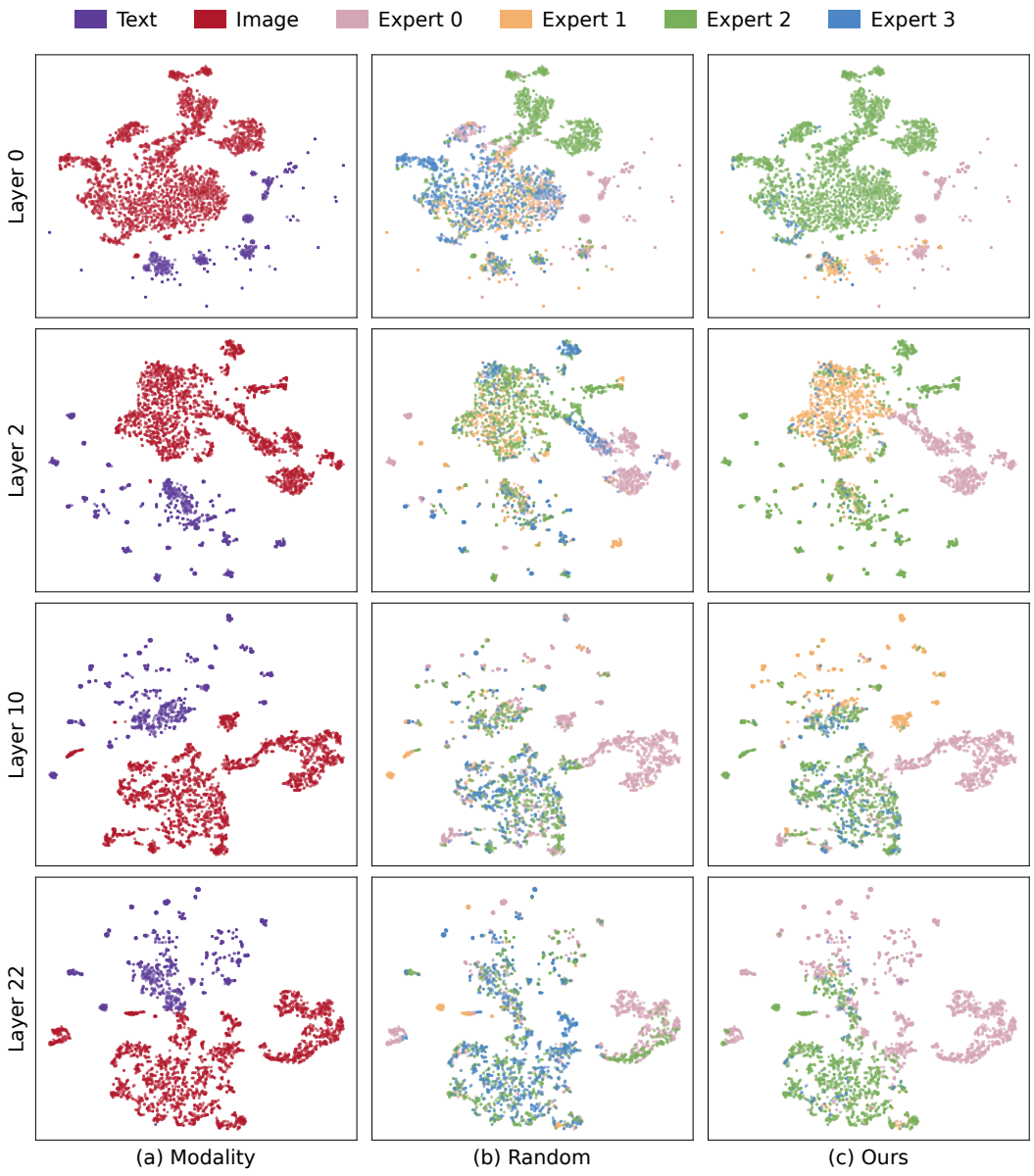

Figure 9: Layer-wise comparison of modality structure (a), random routing (b), and Fisher-initialized routing (c) for **Phi-2 (2.7B)** at layers 0, 2, 10, and 22. The difference between Fisher-initialized and random routing is clearer in earlier layers and becomes less clear at greater depth.

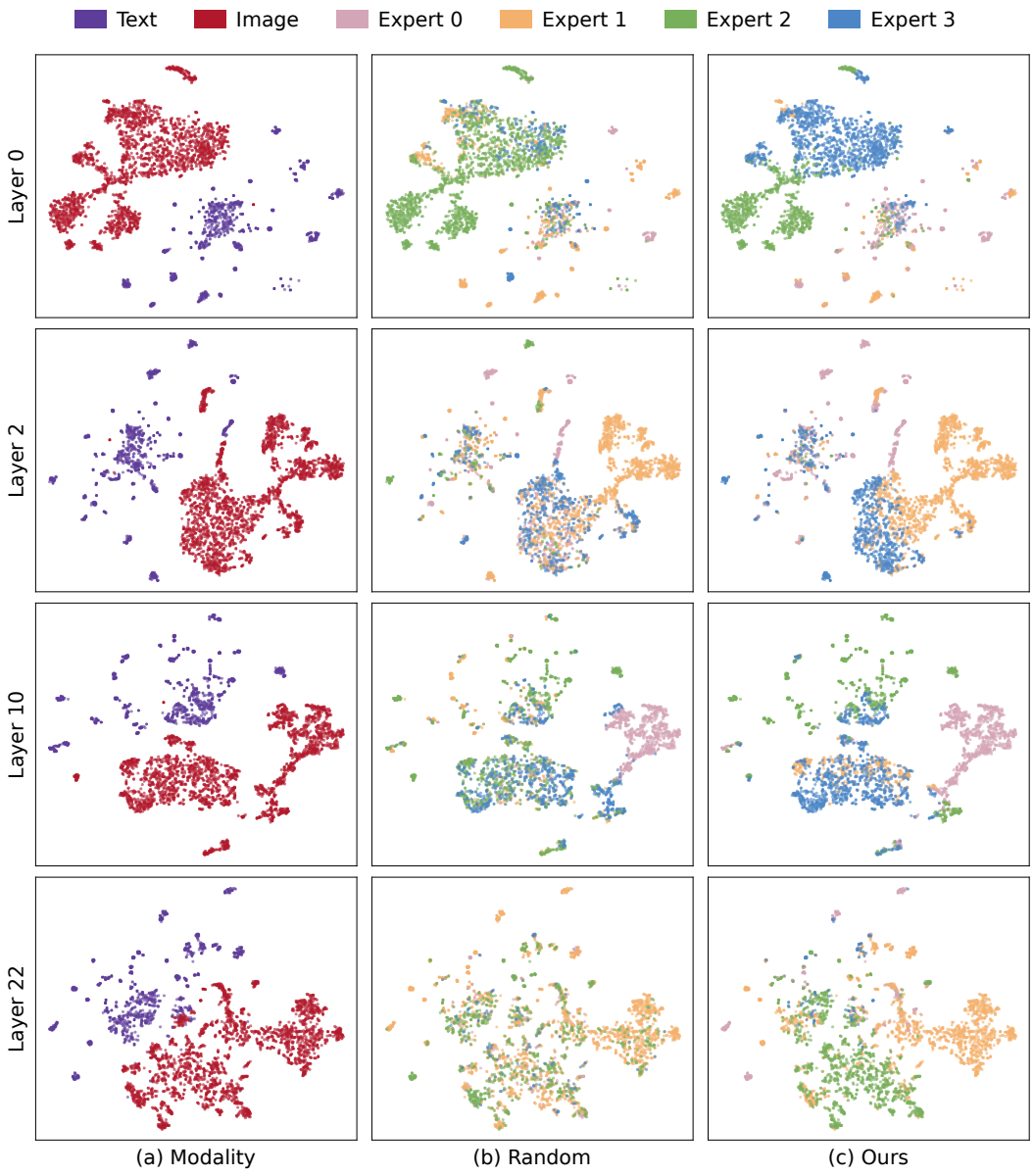

Figure 10: Layer-wise comparison of modality structure (a), random routing (b), and Fisher-initialized routing (c) for **StableLM-1.6B** at layers 0, 2, 10, and 22. As with the other backbones, the difference is more visible in earlier layers and less clear in later layers.