RECONSTRUCTING THE ROUTING SPACE: RETRAINING-FREE COMPRESSION OF VISION-LANGUAGE MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language Mixture-of-Experts models (VLM-MoEs) hold most of their parameters in expert weights, yet their expert pools are hard to compress because visual tokens use experts far more diffusely than text tokens. Our experiments show that covering 90% of visual routing mass takes up to $1.8\times$ as many experts as for text, and that at 50% expert retention frequency-based pruning keeps only 51–53% of dense performance. This exposes a limitation of existing compressors: they rank experts alone or fuse parameters, ignoring the function expressiveness of the routing space. We propose SPARE, which preserves this expressiveness in two steps. It first selects a well-conditioned expert subset by pivoting on the leading eigenvectors of a routing-weighted expert-output Gram matrix, and then reuses these eigenvectors to reconstruct the discarded routing coordinates in closed form. The procedure is one-shot and retraining-free, with no learned parameters. On Kimi-VL-A3B and Qwen3-VL-30B-A3B, SPARE delivers the strongest overall performance across compression budgets, and its advantage widens as compression becomes more aggressive, reaching up to 3.6 points over the strongest baseline at 50% expert retention.

1 INTRODUCTION

Vision-language Mixture-of-Experts models (VLM-MoEs) combine the multimodal reasoning of vision-language models with the conditional computation of sparse expert architectures. Because each token activates only a few experts, MoE layers scale model capacity without a proportional increase in per-token compute. This efficiency comes at a cost in parameters, since every expert must be stored although each token uses only a few, and prior work reports considerable redundancy among experts (Li et al., 2024; Lu et al., 2024; Chen et al., 2024a; Li et al., 2025; Lasby et al., 2025). A large expert pool inflates the memory footprint, hinders deployment on resource-constrained hardware, and adds system overhead, since it fragments token-to-expert assignment and complicates expert scheduling.

Expert pruning and merging have been developed to remove this redundancy in text-only MoEs. Transferring them to VLM-MoEs is not straightforward, because visual tokens route differently from text tokens. Image patches are spatially redundant and individually carry little information (He et al., 2022; Chen et al., 2024b), and they spread routing mass over a markedly larger set of experts: our experiments show that covering 90% of visual routing mass takes up to $1.8\times$ as many experts as covering 90% of textual routing mass (§3.1).

This diffuse routing of visual tokens undermines both families of compressors. Pruning becomes more costly, because many experts contribute comparable shares to the visual representation, so removing any of them perturbs a non-negligible portion of it. Merging, in turn, replaces a group of co-activated experts with a single expert built from their parameters (Li et al., 2024; Chen et al., 2024a; Li et al., 2025). Since an expert's output depends nonlinearly on its parameters, this mixing distorts the function of every member, especially when routing mass is spread evenly across the group. These observations suggest that VLM-MoE compression needs more than discarding rarely used experts or greedily merging similar ones. In particular, a compressor should retain the function expressiveness of the *routing space*—the space of functions a layer realizes by routing over its

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.

entire expert pool—on which multimodal inference relies, and redistribute the routing mass of the discarded experts in a controlled way.

We build on a simple structural fact: although each expert is nonlinear in its parameters, an MoE layer combines the expert outputs linearly, weighted by the router gates. Redundancy is therefore better judged on the routing space, by how much function expressiveness is lost when an expert is removed, than on any single expert's activation statistics. We propose SPARE, which characterizes this space through the experts' outputs on a calibration set, folds routing-weighted importance into their Gram matrix, and selects a compact subset of experts that forms a well-conditioned basis of its principal eigenspace. The same basis then yields a closed-form remapping matrix that expresses the routing mass of every discarded expert as a combination of the retained ones. Unlike merging, this leaves the parameters of the retained experts untouched, so the approximation error is confined to the discarded experts. The whole procedure is one-shot, gradient-free, and introduces no learned parameters.

Our contributions are as follows.

- We measure how visual and text tokens use the expert pool, and show that visual routing is consistently more diffuse across two VLM-MoEs and three corpora (§3).
- We formulate expert compression as preserving the function expressiveness of the routing space, and select the retained experts by pivoting on the principal eigenbasis of a routing-weighted Gram matrix, connecting MoE compression to discrete empirical interpolation (§4.3).
- We derive, from the same basis, a closed-form router remapping that reconstructs the discarded routing coordinates, and thereby the routing space, without retraining.
- On Kimi-VL-A3B and Qwen3-VL-30B-A3B, SPARE delivers the strongest overall performance across compression budgets, with gains that widen under aggressive compression, reaching up to 3.6 points over the strongest baseline at 50% expert retention (§5.2).

2 RELATED WORK

Sparse mixture-of-experts. Sparse mixture-of-experts layers (Shazeer et al., 2017; Fedus et al., 2022) scale capacity by routing each token to a few of N experts, and now underpin both text-only models such as Mixtral (Jiang et al., 2024) and the vision-language models we study. Because the experts hold most of the parameters, they are also where compression pays off. Existing work falls into three largely separate lines.

Expert merging. Merging partitions experts into groups of similar ones and collapses each group into a single module. MC-SMoE (Li et al., 2024) groups by router logits and merges with frequency weighting; HC-SMoE (Chen et al., 2024a) instead clusters hierarchically on expert *outputs*, arguing that router logits are dataset-specific; Sub-MoE (Li et al., 2025) targets parameter conflict by concatenating a cluster's weight matrices, taking one joint SVD, and merging only the private $V^{(i)}$ factors under a shared $U\Sigma$. Whatever signal forms the groups, merging acts on parameters, whereas we preserve the function expressiveness of the routing space and leave the retained parameters untouched. Two properties of this family matter here. First, without rank truncation, Sub-MoE's merge operator reduces algebraically to the frequency-weighted average, $U\Sigma(\sum_n w_n V_n)^\top = \sum_n w_n W_n$ (Appendix B), so differences from HC-SMoE originate in the clustering stage—consistent with the two diverging most where clustering is hardest (§5.2). Second, none of these methods specifies what the *router* should do after merging: the gate still emits N logits while only k modules remain. Our closed-form remapping $\mathbf{M}$ fills this gap.

Expert pruning. Pruning removes experts outright. Criteria include activation frequency, output magnitude, and router-weighted saliency; REAP (Lasby et al., 2025) is a strong recent instance, scoring each expert by its routing mass times its output norm. NAEE (Lu et al., 2024) searches over expert subsets to minimize layerwise reconstruction loss. Scoring criteria of this kind assign each expert a *scalar* and keep the top-k, ranking experts in isolation. This is the assumption our routing-space formulation replaces: a highly scored expert whose output already lies in the span of the retained set is redundant, whereas a rarely routed expert occupying an otherwise uncovered direction is not. A scalar ranking also cannot express that removing two mutually redundant experts is far cheaper than removing two complementary ones.

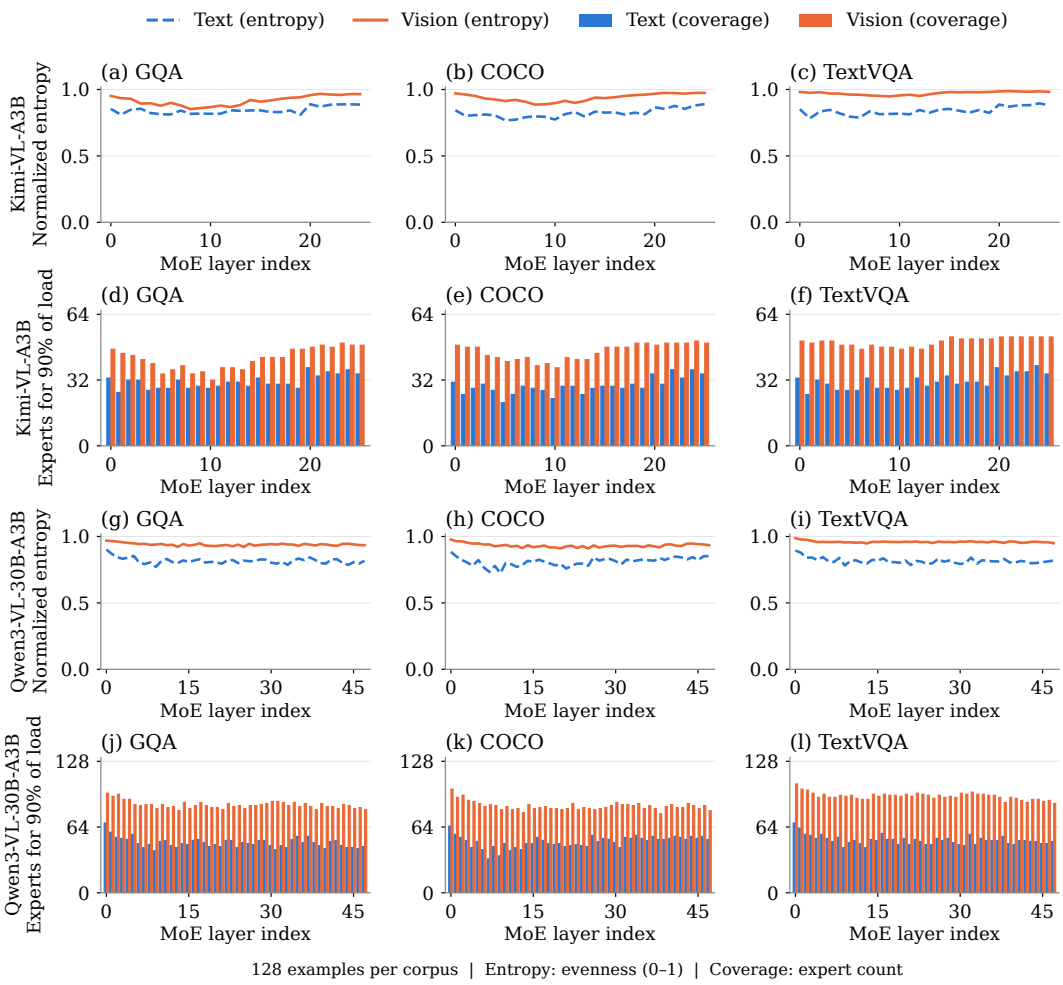


Figure 1: Aggregate routing dispersion on three corpora (128 examples per corpus and model). Columns: GQA, COCO, TextVQA; upper two rows: Kimi-VL, lower two rows: Qwen3-VL. **Lines:** normalized entropy of expert load (0–1) after equal-event multinomial resampling; higher is more even. **Bars:** smallest number of experts covering 90% of routing events, with experts sorted by load in each layer; higher means more experts are needed. Blue: text; orange: vision. Statistics are computed from top-k expert-selection counts.

Dynamic expert skipping. A separate line keeps all weights and skips experts at inference time. MoDES (Huang et al., 2025) is the closest multimodal instance, adaptively skipping experts per token without modifying parameters. Dynamic skipping reduces per-token compute but not the memory footprint, whereas we permanently shrink the expert pool; the two axes are complementary, and MoDES itself names combination with pruning as a natural extension.

3 Motivation

3.1 Visual tokens route more diffusely than text

Existing retraining-free compressors reduce the relations among experts to simple statistics. Pruning criteria rank experts by a scalar computed per expert—activation frequency, output norm, or a router-weighted saliency such as s_j in (2) (Lasby et al., 2025)—and keep the top k. Merging methods group experts by pairwise similarity and fuse the parameters of each group, typically weighting members by activation frequency (Li et al., 2024; Chen et al., 2024a). Both rest on an assumption formed on

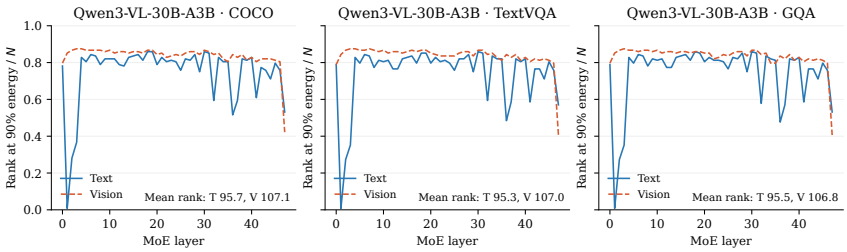


Figure 2: Token-level expert-output rank for Qwen3-VL on three corpora (128 examples each, matched text/vision token counts per example). Columns: COCO, TextVQA, GQA. Blue solid: text; orange dashed: vision. The vertical axis is r_{90}/N, the fraction of spectral directions needed for 90% of the Gram energy; annotations give the unnormalized mean ranks.

text-only MoEs: that routing is concentrated enough for a few experts to dominate, so that a scalar separates load-bearing experts from disposable ones, and each group has a dominant member that can absorb the rest.

Figure 1 compares how visual and text tokens load the experts on GQA, COCO, and TextVQA. On both models and all three corpora, visual tokens spread their load markedly more evenly: the normalized entropy of the expert load is consistently higher, and covering 90% of the routing mass takes substantially more experts—on Qwen3-VL, about two thirds of the entire pool for vision versus well under half for text. Because image tokens far outnumber text tokens, we verify that the gap is not a sample-size artifact by resampling both modalities to equal event counts; the gap is unchanged, so it is a property of the router rather than of the sample (Appendix D.1).

On visual input, a per-expert scalar thus compresses a broad load distribution into a ranking that says nothing about how experts relate to one another, and top-k selection from that ranking can discard complementary experts. The consequence is visible in our main results, where frequency-based pruning, which is determined entirely by this statistic, keeps only about half of dense performance at 50% retention (Table 1). Merging is affected in a different way: with load spread evenly, a group has no dominant member, so fusing its parameters distorts the function of every member, and merging becomes unstable at high compression (§5.2).

Insight 1. On VLM-MoEs, compression can rely neither on a per-expert scalar nor on pairwise similarity. The criterion must judge each expert *against the rest of the pool*—asking what the others can jointly take over—rather than in isolation or in pairs.

3.2 Visual computation occupies a higher-dimensional expert subspace

Routing dispersion describes how experts are used; the rank of their outputs describes the geometry of what they compute. At each MoE layer we evaluate every expert on the same sampled hidden states, stack the outputs of each modality into a matrix $\mathbf{O}$ with one row per expert, whose subspace we call the **expert subspace**, and report r_{90}, the number of leading eigenvalues of $\mathbf{G} = \mathbf{O}\mathbf{O}^\top$ that account for 90% of its trace, i.e., the effective dimension of the expert subspace. Text and vision contribute equally many tokens per example, so the two ranks are directly comparable (details in Appendix D.1).

Figure 2 shows that on VLM-MoEs, vision has the higher rank in the vast majority of layers on all three corpora, needing roughly eleven more directions on average. To retain the same fraction of Gram energy, visual computation therefore spans more of the expert subspace than text, so the choice of retained experts matters more for visual inputs.

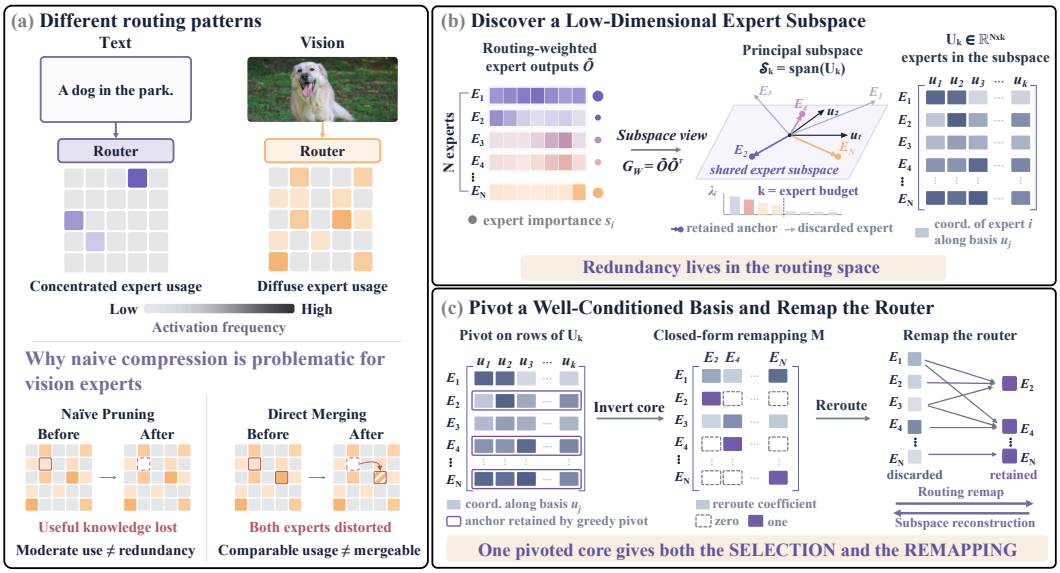


Figure 3: Overview of SPARE. A single calibration pass over 128 multimodal examples records, per MoE layer, the expert-output profile matrix $\mathbf{O}$ and the per-expert importance $\mathbf{s}$. Their routing-weighted Gram matrix $\mathbf{G}_W$ (3) is eigendecomposed once, and greedy pivoting on the rows of its top-k eigenvectors $\mathbf{U}_k$ returns a well-conditioned set $\mathcal{I}$ of retained experts. Inverting that same pivoted core yields the closed-form routing remapping $\mathbf{M} = \mathbf{U}_k \mathbf{U}_k(\mathcal{I}, :)^{-1}$ (5), which folds the routing mass of the discarded experts back onto the retained ones at inference. The pipeline is one-shot and gradient-free, and both stages read from the same spectral object.

Insight 2. Visual tokens draw on a higher-dimensional expert subspace, i.e., visual inference relies on more of the function expressiveness of the routing space. Compression should therefore retain a set of experts that *spans* this subspace well, which is a joint property of the set rather than a sum of per-expert scores or a collection of pairwise similarities.

4 METHOD

Guided by §3, we propose SPARE (**S**ubspace **P**ivoting **A**nd **R**outing r**E**construction), a two-stage pipeline: **(a) joint redundancy-guided selection**, which picks a representative expert subset from a profile matrix that models inter-expert relationships; and **(b) subspace routing reconstruction**, which derives a closed-form remapping $\mathbf{M}$ for the discarded routing coordinates. Both stages serve to preserve the function expressiveness of the routing space and are computed from the same spectral object (Figure 3), so the procedure needs no auxiliary optimization or retraining. Derivations and implementation details are deferred to Appendix A.

4.1 EXPERT REDUNDANCY AS A PROPERTY OF THE ROUTING SPACE

Given an input token $x \in \mathbb{R}^d$, an MoE layer with N experts produces a sparsely weighted sum of expert transformations,

$$y = \sum_{j=1}^{N} g_j(x)\, f_j(x), \tag{1}$$

where $f_j(x) \in \mathbb{R}^d$ is the output of expert j and $g_j(x) \in \mathbb{R}_{\geq 0}$ is its routing weight. The layer output is thus a *linear combination* of expert outputs: the layer realizes its function by routing each token over the experts. We call the space of functions that routing over the whole pool can realize the **routing space**. Its function expressiveness is held by the experts jointly rather than by any one of them, so whether an expert can be removed is decided by whether its removal shrinks this expressiveness, not

by the activation magnitude of that expert alone. This reshapes the notion of redundancy: an expert whose output is well represented by the span of the others can be removed at small reconstruction error even if it is frequently activated, whereas an expert contributing a direction the others do not cover may be indispensable despite a modest scalar score.

4.2 Routing-Aware Expert Geometry

Following Chen et al. (2024a), we characterize each expert by its outputs over a calibration set $\mathcal{D}$, stacked into an **expert profile matrix** $\mathbf{O} \in \mathbb{R}^{N \times (|\mathcal{D}|d)}$ whose i-th row is the function signature of expert i. As in §3.2, we call the subspace of $\mathbf{O}$ the expert subspace; it is the concrete handle through which we characterize the routing space. The Gram matrix $\mathbf{G} = \mathbf{O}\mathbf{O}^\top \in \mathbb{R}^{N \times N}$ carries the linear dependencies among experts at $\mathcal{O}(N^3)$ cost. Following the router-weighted saliency of Lasby et al. (2025), we measure the routing-weighted activation strength of each expert as

$$s_j \;=\; \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} g_j(x)\, \big\| f_j(x) \big\|_2, \qquad \mathbf{s} = (s_1, \ldots, s_N)^\top \in \mathbb{R}^N_{\geq 0}. \tag{2}$$

Geometry alone is not a faithful criterion: an expert may occupy a unique direction yet contribute little if it is rarely routed. We therefore fold $\mathbf{s}$ into the geometry, forming the weighted profile matrix $\widetilde{\mathbf{O}} = \mathrm{diag}(\sqrt{\mathbf{s}})\, \mathbf{O}$ and its Gram matrix

$$\mathbf{G}_W \;=\; \widetilde{\mathbf{O}}\widetilde{\mathbf{O}}^\top \;=\; \mathrm{diag}(\sqrt{\mathbf{s}})\, \mathbf{G}\, \mathrm{diag}(\sqrt{\mathbf{s}}). \tag{3}$$

Because $\mathbf{s}$ enters *before* the spectral decomposition, it reshapes the eigenproblem itself: the leading eigenvectors of $\mathbf{G}_W$ capture the routing-weighted Gram energy of the expert subspace rather than its raw geometric spread. Geometric independence and routing importance thus combine multiplicatively in one spectral object, and an expert is suppressed whenever *either* factor is small — an ordering that a two-step procedure, ranking by $\mathbf{s}$ and then de-duplicating geometrically, cannot reproduce. Ablating the weighting costs up to 9.3 points of MMBench accuracy and 396 MME points, and the damage grows as the budget tightens (Appendix D.2).

4.3 Unified Subspace Selection and Routing Reconstruction

Joint redundancy-guided selection. We recast expert compression as a *subspace coverage problem*: find k experts that span the principal expert subspace of the full pool. Let $\mathbf{U}_k \in \mathbb{R}^{N \times k}$ collect the top-k eigenvectors of $\mathbf{G}_W$, so that $\mathrm{col}(\mathbf{U}_k)$ is the k-dimensional principal expert subspace, which captures the largest share of routing-weighted Gram energy, and the row $\mathbf{U}_k(i,:)$ encodes expert i in this basis. Selecting experts then means choosing a row index set $\mathcal{I} \subset [N]$, $|\mathcal{I}| = k$, that forms a well-conditioned basis of that subspace, which is governed by the volume of the selected rows:

$$\mathcal{I}^\star \;=\; \arg \max_{\mathcal{I} \subset [N],\, |\mathcal{I}| = k} \big| \det \mathbf{U}_k(\mathcal{I}, :) \big|. \tag{4}$$

Algorithm 1 approximates (4) with a greedy pivot that appends, at each step, the expert least explained by those already retained; this is exactly the row that grows the volume the most (Appendix A). It closely tracks the combinatorial optimum (Appendix D.5) and attains the enumerated optimum in every layer of a restricted real-model oracle (Appendix D.6).

Subspace routing reconstruction. The gate still emits N logits while only k experts remain, so we reduce $\mathbf{g} \in \mathbb{R}^N$ through a redistribution matrix $\mathbf{M} \in \mathbb{R}^{N \times k}$, $\tilde{\mathbf{g}} = \mathbf{M}^\top \mathbf{g}$. Requiring that the remapping stay in the principal expert subspace, $\mathrm{col}(\mathbf{M}) \subseteq \mathrm{col}(\mathbf{U}_k)$, and preserve the gate values at the retained experts, $\mathbf{M}(\mathcal{I}, :) = \mathbf{I}_k$, determines $\mathbf{M}$ uniquely (Appendix A):

$$\mathbf{M} \;=\; \mathbf{U}_k\, \mathbf{U}_k(\mathcal{I}, :)^{-1} \;\in\; \mathbb{R}^{N \times k}. \tag{5}$$

Each discarded expert hands its routing mass to the retained experts that express it in the principal expert subspace, so the remapping reconstructs the routing space with k experts and preserves its function expressiveness up to the energy outside that subspace. Selection and reconstruction thus share the same core: the pivots chosen for a well-conditioned expert basis also determine the inverse used by the remapping, so one pass of Algorithm 1 yields the whole pipeline, and pairing $\mathbf{M}$ with any other selection rule destroys the conditioning it relies on (Appendix D.7). No temperature scaling or gradient-based fitting enters the derivation. At inference, the remapped gate passes through the model's existing top-k and normalization steps, and the retained expert parameters are left unchanged.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Setup. We compress two VLM-MoEs with different routing designs: **Kimi-VL-A3B-Instruct** (Kimi Team, 2025) (N=64 routed experts plus 2 shared, top-6 sigmoid routing) and **Qwen3-VL-30B-A3B-Instruct** (Qwen Team, 2025) (N=128, no shared experts, top-8 softmax routing). All statistics are collected in one pass over 128 GQA (Hudson & Manning, 2019) validation examples; no gradient step is taken. We evaluate on TextVQA (Singh et al., 2019), ChartQA (Masry et al., 2022), MMStar (Chen et al., 2024c), MMBench-EN-dev (Liu et al., 2024), MME (Fu et al., 2023), and POPE (Li et al., 2023) with `lmms-eval` (Zhang et al., 2025); the aggregate (Avg.) is the mean of the six scores, with MME divided by its maximum of 2800, reported as a percentage of the dense model. Architecture, calibration and implementation details are in Appendix C.

Baselines. We compare against two merging methods, **HC-SMoE** (Chen et al., 2024a) and **Sub-MoE** (Li et al., 2025), and two pruning methods, **REAP** (Lasby et al., 2025) and **Freq-Prune**, which drops the least frequently activated experts. None reports VLM-MoE results, so all are re-run under our protocol with the same calibration data and the same number of retained experts k, making every block of our tables iso-parameter.

5.2 MAIN RESULTS

Table 1 reports the six benchmarks and the aggregate for both models at three compression budgets, and SPARE achieves the highest aggregate score in every block (shared with Sub-MoE on Kimi-VL at 75%). The margin grows with the compression ratio. At the mild 90% budget every method except Freq-Prune lies within about one point of the others, since removing a tenth of the experts is easy for any reasonable rule; at 50%, SPARE leads the strongest baseline by 1.65 points on Kimi-VL (89.29 vs. 87.64) and by 3.61 points on Qwen3-VL (91.24 vs. 87.63). This is the regime the routing-space view predicts: as k approaches the effective rank of the expert subspace, *which* experts are retained decides whether the layer's function can still be reconstructed, and keeping the individually highest-scoring experts no longer suffices.

The two baseline families fail in different ways. Freq-Prune is the weakest throughout and collapses rather than degrading gracefully, retaining only 50.73% (Kimi-VL) and 52.78% (Qwen3-VL) of dense performance at 50%, with the loss concentrated on individual benchmarks—its Kimi-VL MMBench score falls to 6.35 while POPE stays at 82.21. This is what §3.1 predicts: once visual tokens spread routing mass broadly, activation counts stop separating load-bearing experts from interchangeable ones. Merging is unstable at high compression, where Sub-MoE drops to 69.53% on Kimi-VL, 14.88 points *below* plain frequency-weighted averaging (HC-SMoE, 84.41%); because the two share an algebraically equivalent merge operator (Appendix B), the gap stems from clustering, which degrades sharply once clusters must absorb many dissimilar experts, whose fused parameters distort the function of every member, as §3.1 anticipates. SPARE sidesteps both clustering and parameter fusion, and Appendix D.8 shows TextVQA examples at 50% retention where it preserves the answer while both pruning baselines do not.

5.3 PUSHING THE COMPRESSION RATIO

Deployment budgets are often set by the hardware rather than by the model, and a VLM-MoE that does not fit is of no use at any accuracy. SPARE makes the expert pool small enough to matter in that regime. Keeping 37.5% of the experts (Table 2) removes 9.0 B parameters from Kimi-VL and 18.1 B from Qwen3-VL, roughly 18 and 36 GB of bfloat16 weights, and the compressed models remain usable: they answer with 80.4% and 85.0% of the dense average, leading REAP on both models (78.3% and 82.7%), on four of the six benchmarks on Kimi-VL and on all six on Qwen3-VL. Capability survives where it is needed: ChartQA stays at 66.0 and 58.5 and MME at 1731 and 1971 out of 2800. A practitioner constrained to a smaller accelerator can therefore still run these models with most of their multimodal ability intact, which is exactly the setting where the choice of retained experts stops being a matter of a few points and decides whether the layer works at all.

Table 1: Expert compression on VLM-MoEs, calibrated on GQA (128 samples). **Avg.** is the mean of the six benchmark scores after rescaling MME by its maximum of 2800, reported as a percentage of the dense model's average. Best per block in **bold**.

Method	#E	TextVQA	ChartQA	MMStar	MMBench	MME	POPE	Avg.(%)
Kimi-VL								
Dense (uncompressed)	64	88.53	85.64	50.09	82.56	2249	87.46	100.00
Keep 90% of experts (N=64 → k=58)								
HC-SMoE	58	81.93	84.80	50.83	81.30	2174	87.77	97.82
Sub-MoE	58	81.44	84.76	51.04	81.52	2143	87.90	97.60
REAP	58	81.74	84.48	48.55	82.04	2137	87.57	97.06
Freq-Prune	58	80.63	73.36	50.63	80.32	2131	86.64	94.32
SPARE (Ours)	58	82.52	85.16	49.31	81.87	2193	87.69	**97.94**
Keep 75% of experts (N=64 → k=48)								
HC-SMoE	48	79.15	81.84	51.71	77.40	2069	87.21	95.06
Sub-MoE	48	79.20	80.72	58.30	77.57	2185	87.84	**97.27**
REAP	48	80.64	78.88	47.33	80.15	2145	87.61	95.07
Freq-Prune	48	79.02	68.52	49.95	77.83	2083	78.12	90.14
SPARE (Ours)	48	80.85	84.12	51.70	80.07	2157	87.89	**97.27**
Keep 50% of experts (N=64 → k=32)								
HC-SMoE	32	69.63	73.72	46.42	61.76	1721	87.62	84.41
Sub-MoE	32	64.77	59.08	37.87	39.26	1167	87.33	69.53
REAP	32	74.54	62.20	51.33	71.47	1970	86.06	87.64
Freq-Prune	32	43.59	26.64	27.59	6.35	1523	82.21	50.73
SPARE (Ours)	32	75.58	73.44	47.67	72.25	1857	88.50	**89.29**
Qwen3-VL								
Dense (uncompressed)	128	83.74	86.16	59.51	86.60	2486	90.01	100.00
Keep 90% of experts (N=128 → k=115)								
HC-SMoE	115	83.32	86.24	60.44	85.91	2320	90.10	98.80
Sub-MoE	115	83.68	86.32	59.26	85.48	2351	90.13	98.80
REAP	115	83.55	85.12	60.41	86.51	2404	90.03	99.33
Freq-Prune	115	81.91	70.68	56.88	86.16	2347	90.32	94.94
SPARE (Ours)	115	83.45	86.88	60.93	87.03	2416	90.04	**99.96**
Keep 75% of experts (N=128 → k=96)								
HC-SMoE	96	81.71	86.32	61.15	82.81	2126	89.69	96.53
Sub-MoE	96	81.98	86.64	60.51	83.07	2101	89.80	96.41
REAP	96	82.29	72.92	63.30	85.65	2327	90.37	96.53
Freq-Prune	96	77.06	57.40	37.52	79.72	2301	90.31	85.73
SPARE (Ours)	96	82.76	86.76	61.09	84.71	2292	90.34	**98.53**
Keep 50% of experts (N=128 → k=64)								
HC-SMoE	64	77.22	79.92	47.95	67.35	1935	86.99	86.61
Sub-MoE	64	77.36	80.40	52.12	67.52	1913	87.13	87.48
REAP	64	80.38	54.44	54.69	81.79	2033	89.71	87.63
Freq-Prune	64	18.62	30.88	10.87	57.30	1840	77.77	52.78
SPARE (Ours)	64	79.91	65.84	60.92	81.87	2051	89.66	**91.24**

Table 2: Compression to 37.5% of the experts, past the tightest budget of Table 1: k=24 for Kimi-VL and k=48 for Qwen3-VL. Avg. relative to dense.

Model	Method	TextVQA	ChartQA	MMStar	MMBench	MME	POPE	Avg.(%)
Kimi-VL (k=24)	SPARE	69.76	65.96	44.77	62.97	1731	86.67	80.36
	REAP	69.17	57.36	44.52	63.57	1757	84.50	78.29
Qwen3-VL (k=48)	SPARE	78.50	58.48	52.31	75.26	1971	89.57	85.04
	REAP	75.38	55.44	50.51	73.63	1954	87.87	82.66

The pivot selects what an exhaustive search would. The selection rule can be tested directly against enumeration. On the Kimi-VL calibration geometry we enumerate all $\binom{12}{6} = 924$ subsets of a restricted candidate pool in every MoE layer and score them with the weighted maximum-volume objective of (4). SPARE attains the enumerated optimum in all 26 layers and frequency pruning in none, and the mean Lebesgue constant—the quantity that bounds the reconstruction error of the

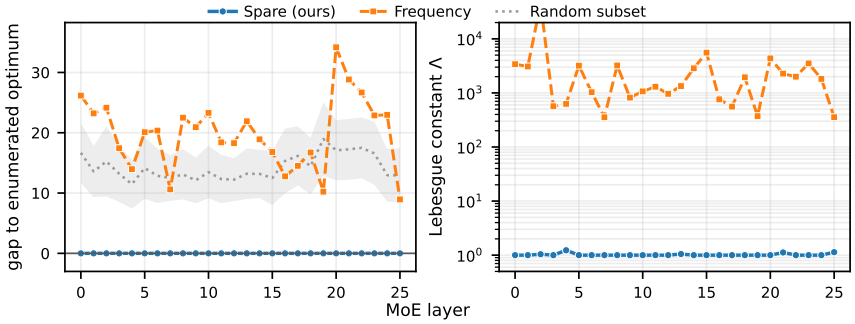


Figure 4: Restricted real-model oracle on the Kimi-VL calibration geometry: all $\binom{12}{6} = 924$ subsets of a 12-expert candidate pool are enumerated in each of the 26 MoE layers and scored with the weighted maximum-volume objective (4). **Left:** gap to the enumerated optimum; zero means the method selects the optimal subset, which SPARE does in all 26 layers. The shaded band is the standard deviation of random subsets of the same pool. **Right:** Lebesgue constant of the selected core, which bounds the reconstruction error of the remapping (log scale).

remapping—is 1.02 for SPARE against 3551.92 (Figure 4). Frequency ranking is in fact worse than picking experts at random from the same pool in 21 of the 26 layers, which is the geometric statement behind its collapse in Table 1. The greedy pivot is therefore not a cheap heuristic: it recovers the basis the objective asks for, which is why the router handed to (5) stays well conditioned where a frequency ranking leaves it three orders of magnitude worse (Appendix D.6).

5.4 ROBUSTNESS TO THE CALIBRATION SET

The selection is stable across calibration corpora and sizes. At the tightest budgets, expert sets selected from GQA, COCO, and TextVQA overlap by 86–93%, 36–43 percentage points above the overlap of random sets. Recompressing Kimi-VL at k=48 from each corpus changes the aggregate by at most 0.33 points, and varying the calibration size from 32 to 512 examples moves it by under one point on either model (Appendix D.3).

5.5 EFFICIENCY

Compressing to budget k removes $(N-k)/N$ of the routed-expert parameters: 7.20 B of Kimi-VL's 14.39 B and 14.50 B of Qwen3-VL's 28.99 B at the 50% budget. The router still activates k_{top} experts per token, so the activated parameters per token are preserved and the savings go entirely to the memory and storage footprint, the main obstacle to deploying large MoEs on resource-constrained hardware; methods that lower k_{top}, such as dynamic expert skipping (Huang et al., 2025), act on the complementary axis and compose with ours. Compression itself is one-shot and gradient-free, dominated by a single calibration pass, with only an $N \times N$ eigendecomposition and a $k \times k$ inverse per layer afterwards (Appendix C).

6 CONCLUSION

We revisited retraining-free expert compression for vision-language MoEs and identified a measurable reason why ranking experts alone or fusing their parameters becomes unreliable: visual tokens spread routing mass over more experts, and occupy more directions of the expert subspace, than text tokens. SPARE instead preserves the function expressiveness of the routing space: from a single calibration pass, it selects a well-conditioned basis of the expert subspace by pivoting on a routing-weighted Gram matrix and reuses that basis to remap the router in closed form, with no learned parameters and no retraining. It delivers the strongest overall performance on both evaluated VLM-MoEs, leading the strongest baseline by up to 3.6 points under the most aggressive compression. Because it leaves the number of activated experts per token unchanged, it composes directly with dynamic expert skipping and extends naturally to joint compression with visual-token reduction.

AI DISCLOSURE

In this work, we used generative AI tools to aid and polish the writing of the paper, including improving readability, wording, and the organization of the text, and for retrieval and discovery, namely identifying and locating related work. We have not used generative AI tools for generating synthetic data, developing theoretical models or frameworks, formulating mathematical claims or proofs, proposing or refining hypotheses, designing the research methodology or experiments, implementing the methods, translation, qualitative data analysis, or result interpretation; dataset cleaning and reformatting are not applicable to this work. We have reviewed all AI-assisted work: every AI-suggested edit was checked by the authors against the experimental records to ensure that no reported number or claim was altered, and every related work surfaced with AI assistance was read and verified by the authors before being cited. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010. doi: 10.1137/090766498.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse mixture-of-experts via hierarchical clustering. *arXiv preprint arXiv:2410.08589*, 2024a. URL https://arxiv.org/abs/2410.08589.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision (ECCV)*, 2024b.

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? *arXiv preprint arXiv:2403.20330*, 2024c.

Zlatko Drmač and Serkan Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016. doi: 10.1137/15M1019271.

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.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yushi Huang, Zining Wang, Zhihang Yuan, Ruihao Gong, Yifu Ding, Jinyang Guo, Xianglong Liu, and Jun Zhang. MoDES: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping. *arXiv preprint arXiv:2511.15690*, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Kimi Team. Kimi-VL technical report. *arXiv preprint arXiv:2504.07491*, 2025. URL https://arxiv.org/abs/2504.07491.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot mixture-of-experts compression. *arXiv preprint arXiv:2510.13999*, 2025. URL https://arxiv.org/abs/2510.13999.

Lujun Li, Qiyuan Zhu, Jiacheng Wang, Wei Li, Hao Gu, Sirui Han, and Yike Guo. Sub-MoE: Efficient mixture-of-expert LLMs compression via subspace expert merging. *arXiv preprint arXiv:2506.23266*, 2025. URL https://arxiv.org/abs/2506.23266.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy. In *International Conference on Learning Representations (ICLR)*, 2024.

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, 2023. URL https://aclanthology.org/2023.emnlp-main.20.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755. Springer, 2014. URL https://doi.org/10.1007/978-3-319-10602-1_48.

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? *arXiv preprint arXiv:2307.06281*, 2024.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

Qwen Team. Qwen3-VL-MoE. Model release, Hugging Face, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

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

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326, 2019.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.

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

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-naacl.51/.

A METHOD DERIVATIONS

Why the Gram matrix. The left singular subspace of the profile matrix $\mathbf{O}$ exposes the linear dependencies among experts, but with an effective feature dimension $|\mathcal{D}|d \gg N$ a direct SVD of $\mathbf{O}$ is prohibitive. The Gram matrix $\mathbf{G} = \mathbf{O}\mathbf{O}^\top \in \mathbb{R}^{N\times N}$ shares the left singular vectors of $\mathbf{O}$ and has the squared singular values as its eigenvalues, so it carries the same subspace information at the cost of one $\mathcal{O}(N^3)$ eigendecomposition of an $N \times N$ matrix. It is accumulated in a single calibration pass (Appendix C).

Algorithm 1 Greedy Pivoting for Expert Selection

Require: Top-k eigenvectors $\mathbf{U}_k \in \mathbb{R}^{N\times k}$
Ensure: Retained index set $\mathcal{I} \subset \{1, \dots, N\}$, $|\mathcal{I}| = k$
$i_1 \leftarrow \arg\max_{i\in[N]} |\mathbf{U}_k(i, 1)|$
$\mathcal{I} \leftarrow \{i_1\}$
for $j = 2, \dots, k$ **do**
  $\mathbf{A} \leftarrow \mathbf{U}_k(\mathcal{I}, 1{:}j-1)$ ▷ $(j-1)\times(j-1)$ core block
  $\mathbf{b} \leftarrow \mathbf{U}_k(\mathcal{I}, j)$
  $\mathbf{r}_j \leftarrow \mathbf{U}_k(:, j) - \mathbf{U}_k(:, 1{:}j-1)\,\mathbf{A}^{-1}\mathbf{b}$ ▷ residual along direction j
  $i_j \leftarrow \arg\max_{i\notin\mathcal{I}} |\mathbf{r}_j(i)|$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_j\}$
end for
return $\mathcal{I}$

Why the top-k eigenvectors. By the Courant–Fischer characterization, the span of the top-k eigenvectors of $\mathbf{G}_W$ is the k-dimensional principal expert subspace, which captures the largest share of routing-weighted Gram energy; this is why selection in §4.3 operates on the rows of $\mathbf{U}_k$. A well-conditioned row subset is one of large volume, and the core $\mathbf{U}_k(\mathcal{I}, :)$ returned by Algorithm 1 is nonsingular in exact arithmetic, so the remapping (5) is always defined.

The greedy step maximizes volume growth. At step j, Algorithm 1 forms the residual $\mathbf{r}_j$ of the j-th principal direction $\mathbf{u}_j$ against the span of the rows already retained and appends

$$i_j \;=\; \arg\max_{i\notin\mathcal{I}_{j-1}} \big|\mathbf{r}_j(i)\big|, \qquad \big|\mathbf{r}_j(i)\big| \;=\; \frac{\big|\det \mathbf{U}_k(\mathcal{I}_{j-1}\cup\{i\},\, 1{:}j)\big|}{\big|\det \mathbf{U}_k(\mathcal{I}_{j-1},\, 1{:}j{-}1)\big|}. \tag{6}$$

The right-hand identity is the Schur-complement expansion of the $j \times j$ determinant along its last row: the residual entry $|\mathbf{r}_j(i)|$ is exactly the multiplicative growth of the volume of the selected block when row i is appended. Maximizing it is therefore the greedy step of the volume objective (4), and it adds the expert whose component along $\mathbf{u}_j$ is least explained by the experts already retained. The procedure is a greedy approximation; it is not claimed to solve (4) globally (Appendix D.5).

Closed form of the remapping. The first requirement on $\mathbf{M}$, $\mathrm{col}(\mathbf{M}) \subseteq \mathrm{col}(\mathbf{U}_k)$, forces $\mathbf{M} = \mathbf{U}_k\mathbf{C}$ for some $\mathbf{C} \in \mathbb{R}^{k\times k}$. Substituting this into the second, the interpolation condition $\mathbf{M}(\mathcal{I}, :) = \mathbf{I}_k$, gives

$$\mathbf{U}_k(\mathcal{I}, :)\,\mathbf{C} \;=\; \mathbf{I}_k, \tag{7}$$

whose coefficient matrix is the core block selected by Algorithm 1. The pivoting rule keeps that core well conditioned, so the system has the unique solution $\mathbf{C} = \mathbf{U}_k(\mathcal{I}, :)^{-1}$ and hence (5). The operator $\mathbf{U}_k(\mathbf{P}^\top\mathbf{U}_k)^{-1}\mathbf{P}^\top$ with $\mathbf{P}$ the selection matrix of $\mathcal{I}$ is the oblique projection of discrete empirical interpolation Chaturantabut & Sorensen (2010); Drmač & Gugercin (2016); here it acts on routing coordinates rather than on sampled spatial points.

The importance score $\mathbf{s}$. Equation (2) uses $s_j = \frac{1}{|\mathcal{D}|}\sum_x g_j(x)\|f_j(x)\|_2$, where $g_j(x)$ is the renormalized top-k routing weight actually applied to expert j in the forward pass. REAP's saliency differs in two details: a softmax over all N experts before restricting to the top-k mass, and padding positions masked out. We keep the simpler form because $\mathbf{s}$ enters the method only through the symmetric scaling in (3), which is invariant to global positive rescaling; what matters is the relative magnitude across experts within a layer. Any non-negative per-expert importance can be substituted. REAP is run with its own score unchanged, so the comparison in Table 1 is unaffected.

B SUB-MOE'S MERGE OPERATOR WITHOUT TRUNCATION

Without rank truncation, Sub-MoE's joint-SVD merge coincides exactly with the frequency-weighted average, a fact used in §2 and §5.2; the difference between Sub-MoE and HC-SMoE in Table 1 therefore lies entirely in the clustering stage.

Let $W_1, \ldots, W_N$ be the expert weight matrices of one cluster and let $w_1, \ldots, w_N$ be normalized activation frequencies, $\sum_n w_n = 1$. Sub-MoE concatenates the experts and takes one joint SVD,

$$[\, W_1 \; \cdots \; W_N \,] \;=\; U\Sigma\, [\, V_1 \; \cdots \; V_N \,]^\top, \tag{8}$$

so that each expert factors through the shared left factor as $W_n = U\Sigma V_n^\top$. The merged expert is formed by averaging the private right factors and reconstructing; since $U\Sigma$ does not depend on n, the reconstruction is linear in the V_n:

$$W_{\text{merged}} = U\Sigma\Big(\sum_n w_n V_n\Big)^\top = \sum_n w_n\, U\Sigma V_n^\top = \sum_n w_n W_n. \tag{9}$$

The step requires only that the factorization be exact. The Sub-MoE configuration we compare against uses `full_matrices=False` with no rank truncation, so the identity holds; verifying numerically (random instances, $N = 4$, $d_{\text{ff}} = 128$, $d_{\text{model}} = 64$), the maximum absolute deviation is 2.74×10^{-6} with relative error 1.25×10^{-6}, i.e. float32 SVD round-trip noise. Consequently, any difference between Sub-MoE and HC-SMoE comes from clustering (K-means on expert-output cosine similarity versus hierarchical), consistent with their divergence on Kimi-VL at high compression (§5.2), and Table 1 pairs each merging operator with the clustering procedure its authors specify.

The truncated variant Sub-MoE† adds activation-aware whitening in the manner of Wang et al. (2024) and truncates Σ, which breaks the identity; it compresses *within* experts rather than reducing the expert count, a different axis from the one studied here.

C ADDITIONAL EXPERIMENTAL DETAILS

Models. The two VLM-MoEs differ in expert granularity, routing rule, and the presence of shared experts, so that our conclusions are not tied to one architecture. Kimi-VL-A3B-Instruct has 27 layers with N=64 routed experts plus 2 shared experts per layer, activates 6 experts per token, and routes with a sigmoid scoring function under the `noaux_tc` grouped-selection rule; its first layer is dense (`first_k_dense_replace`= 1) and is left untouched. Qwen3-VL-30B-A3B-Instruct has 48 layers with N=128 routed experts and no shared experts, activates 8 experts per token, and uses standard softmax top-k routing.

Calibration data. Unless stated otherwise, statistics are collected on 128 randomly sampled GQA validation examples, the only data the method ever sees; no benchmark training data is used. §5.4 varies this choice with COCO (Lin et al., 2014) and TextVQA. The text-only model (Appendix D.4) is calibrated on 128 C4 (Raffel et al., 2020) documents of 2048 tokens.

Benchmarks and aggregate. The six benchmarks cover OCR and text reading (TextVQA), chart reasoning (ChartQA), general multimodal reasoning (MMStar, MMBench), perception (MME), and hallucination (POPE). MME is reported on its native 2800-point scale and divided by 2800 for the aggregate so that it is commensurate with the other five metrics.

Baselines. HC-SMoE merges experts by hierarchical clustering on expert outputs with frequency-weighted averaging. Sub-MoE clusters with K-means on expert-output cosine similarity and merges through a joint SVD with a shared $U\Sigma$ factor. REAP prunes by a router-weighted saliency score. Freq-Prune counts activations over both text and visual tokens; it is the natural multimodal transfer of frequency-based criteria and serves as the naive anchor of each block. All methods use the same calibration set and size, compression ratio, and evaluation harness, and the pruning methods keep the model's original top-k activation rule.

Hardware. All runs use bfloat16 weights with FlashAttention-2 on a single H20 (96 GB).

Iso-cost comparison. SPARE, REAP, and Freq-Prune are static structural pruning at a fixed k and produce architecturally identical models—the same number of experts, the same k_{top}, the same tensor shapes—so each block of Table 1 is an iso-parameter comparison by construction. The merging baselines are iso-parameter after de-duplication but retain N routing slots in their implementations, which if anything favours them.

Statistics collection and offline cost. One calibration pass records, per MoE layer, the expert-output Gram $\mathbf{G}$, the importance vector $\mathbf{s}$, and per-modality routing counts. This pass dominates the cost of compression: $\mathbf{G}$ requires evaluating every expert on every token, so it costs roughly N/k_{top} normal forwards—$11\times$ for Kimi-VL (top-6 of 64) and $16\times$ for Qwen3-VL (top-8 of 128). It is paid once per (model, calibration set) pair rather than once per compressed checkpoint, because the resulting tensors are small ($26 \times 64 \times 64$ and $48 \times 128 \times 128$ fp32, 0.4 MB and 3.1 MB) and are reused across every budget and ablation in this paper. The post-calibration pipeline (weighting, eigendecomposition, greedy pivot, $k \times k$ inverse) takes 2.3 ms per Kimi-VL layer and 7.6 ms per Qwen3-VL layer on CPU, i.e. well under a second for either model.

Decoding and evaluation. All benchmarks run through `lmms-eval` with each task's default prompt and greedy decoding, identical settings across methods except the compression applied. MME is the sum of the perception and cognition sub-scores.

Reproducibility notes. Calibration sampling uses seed 42. Evaluation data are served from a pre-populated offline cache, so no benchmark can change within a sweep. Generation errors abort the run rather than being scored as empty answers.

D ADDITIONAL EXPERIMENTS

D.1 ROUTING AND RANK STATISTICS

This section reports the numbers behind §3; all statistics use 128 examples per corpus.

Routing dispersion. Table 3 summarizes GQA, averaged over layers. Visual tokens have higher normalized load entropy, a lower Gini coefficient, and need more experts to cover 90% of the routing mass on both models. COCO and TextVQA show the same direction: the visual–text entropy gaps are 0.117 and 0.133 on Kimi-VL and 0.122 and 0.143 on Qwen3-VL, and vision needs more experts for 90% coverage in all four settings.

Table 3: Aggregate routing dispersion on GQA, averaged over MoE layers.

	Normalized entropy ↑		Experts for 90% mass		Gini	
Model	Text	Vision	Text	Vision	Text	Vision
Kimi-VL (N=64)	0.843	0.916	31.1	42.4	0.597	0.441
Qwen3-VL (N=128)	0.817	0.940	48.8	85.7	0.683	0.411

Controlling for token count. On GQA, image tokens outnumber text tokens by $9\times$ (Kimi-VL) and $13.5\times$ (Qwen3-VL), and plug-in entropy is biased downward at small sample sizes. We therefore resample each modality's load distribution to the smaller event count with 200 multinomial draws. The entropy gap is unchanged—$+0.0727$ rarefied versus $+0.0727$ plug-in on Kimi-VL, and $+0.1233$ versus $+0.1235$ on Qwen3-VL—and the Miller–Madow correction agrees.

Expert-output rank. The columns of $\mathbf{O}$ are token–hidden-coordinate pairs, and $\mathbf{G}$ is accumulated uncentered. Within each example we sample $\min(32, T_{\text{text}}, T_{\text{vision}})$ tokens per modality without replacement and use the same positions at every layer; the forward pass uses the full input, and individual token outputs enter $\mathbf{G}$ without averaging. On Qwen3-VL, mean r_{90} rises from 95.69 (text) to 107.06 (vision) on COCO, from 95.29 to 107.04 on TextVQA, and from 95.46 to 106.75 on GQA; vision has the higher rank in 44, 45, and 42 of the 48 layers, with median paired differences of 4.5, 5.0, and 4.5 directions.

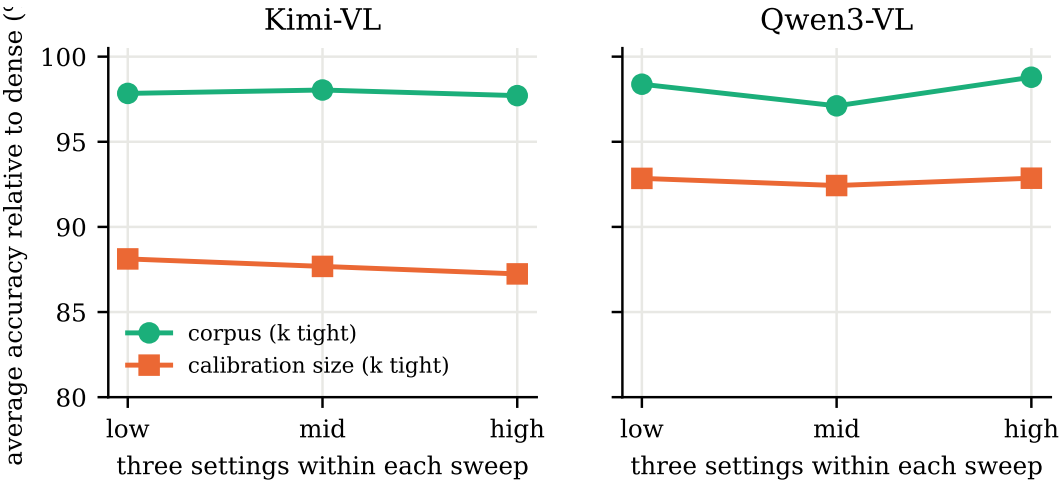


Figure 5: Calibration robustness across two VLM-MoEs, three calibration corpora, and three calibration-set sizes. Each panel contains six measured settings; the variation is small compared with the effect of the compression budget.

D.2 Complete Component Ablation

Table 4 ablates both components of SPARE at every budget, including the two budgets below the main sweep (k=16 on Kimi-VL and k=32 on Qwen3-VL). This sweep carries its own dense reference, so its scores are comparable within the table rather than against Table 1. Both components help almost everywhere: removing the remapping lowers 46 of the 48 benchmark scores, and removing the importance weighting lowers every score at $k \leq 48$ on Kimi-VL and $k \leq 96$ on Qwen3-VL. The few exceptions sit at the loosest budget of each model (k=58 and k=115), where discarding a tenth of the experts barely moves any score and the variants have little left to separate. Both effects also grow as the budget tightens: the remapping is worth 0.3 MMBench points at k=115 but 3.5 at k=64 on Qwen3-VL, and 0.1 at k=58 but 4.0 at k=16 on Kimi-VL, while dropping the weighting costs 396 MME points at k=32 on Kimi-VL (1857 $\rightarrow$ 1461) and 9.3 MMBench points at k=64 on Qwen3-VL (82.99 $\rightarrow$ 73.71).

D.3 Calibration Corpus and Size

Table 5 and Figure 5 give the accuracy counterpart of the set-overlap analysis in §5.4. Recompressing from COCO or TextVQA instead of GQA moves the aggregate by at most 1.3 points (Qwen3-VL, COCO) and by at most 0.2 points on Kimi-VL. Varying the number of calibration examples over n=32–512 at the tightest budget of each model (k=32 and k=64) moves it by 0.88 points on Kimi-VL and 0.43 on Qwen3-VL, with no trend consistent across the two models. Both checks point to the same conclusion: what $\mathbf{G}_W$ contributes is the *identity* of the directions it measures, which 32–128 examples already estimate stably, rather than fine details of its spectrum.

D.4 Text-Only Transfer

To test generality beyond the multimodal setting, we apply the same code and calibration procedure (128 C4 documents) to the text-only **Qwen3-30B-A3B** (Yang et al., 2025) and evaluate on the eight lm-eval tasks of the Sub-MoE protocol at k=96 (75%) and k=64 (50%). Table 6 reports the results. SPARE transfers without code changes, retaining 95.9% of dense performance at k=96 and 83.3% at k=64. It outperforms both merging baselines by 27 and 34 points of retained performance, and matches REAP to within a quarter of a point of average accuracy at either budget (69.38 vs. 69.60 and 60.20 vs. 60.45), leading it on three of the eight tasks in each case. This is what §3 predicts: the margin that subspace selection buys over a scalar saliency score is a multimodal effect, which appears where routing is diffuse rather than on text-only input, where the two criteria order the experts almost identically.

Table 4: Complete ablation of both SPARE components at all budgets, including the two budgets beyond the main sweep. Rows come from one sweep and are directly comparable to one another. The *w/o importance weighting* row at k=16 omits MMBench, whose score is a dynamic LLM judge frozen under the rule-scored protocol of this sweep.

Variant	TextVQA	ChartQA	MMStar	MMBench	MME	POPE
Kimi-VL						
Dense (uncompressed)	88.37	88.72	61.06	82.22	2240	87.38
N=64 → k=58						
SPARE (full)	87.89	88.80	60.87	82.65	2205	87.31
w/o importance weighting	87.74	88.84	61.43	81.62	2208	87.68
w/o routing reconstruction ($\mathbf{M}$)	88.05	88.76	60.30	82.52	2199	87.20
N=64 → k=48						
SPARE (full)	85.34	87.92	59.36	79.98	2153	87.72
w/o importance weighting	83.72	86.52	56.02	79.90	1985	86.83
w/o routing reconstruction ($\mathbf{M}$)	85.05	87.08	58.11	79.30	2116	86.92
N=64 → k=32						
SPARE (full)	78.24	77.88	45.99	71.56	1857	87.66
w/o importance weighting	74.27	74.44	45.50	65.12	1461	85.32
w/o routing reconstruction ($\mathbf{M}$)	77.56	77.16	45.10	68.47	1775	86.73
N=64 → k=16						
SPARE (full)	55.92	52.52	37.10	45.70	1436	84.06
w/o importance weighting	49.03	50.20	33.66	–	1327	82.04
w/o routing reconstruction ($\mathbf{M}$)	50.61	51.28	36.63	41.75	1331	83.26
Qwen3-VL						
Dense (uncompressed)	83.79	86.04	63.39	87.11	2487	90.02
N=128 → k=115						
SPARE (full)	83.52	87.16	64.90	86.68	2431	89.99
w/o importance weighting	83.04	85.48	64.03	86.17	2364	90.10
w/o routing reconstruction ($\mathbf{M}$)	83.09	87.00	64.41	86.34	2421	90.10
N=128 → k=96						
SPARE (full)	82.49	85.04	65.21	85.14	2326	90.13
w/o importance weighting	81.17	84.60	63.71	84.71	2192	89.36
w/o routing reconstruction ($\mathbf{M}$)	82.21	84.48	64.62	84.88	2306	89.22
N=128 → k=64						
SPARE (full)	81.07	70.40	63.06	82.99	2073	89.81
w/o importance weighting	77.77	69.68	57.22	73.71	2028	89.77
w/o routing reconstruction ($\mathbf{M}$)	80.91	69.28	60.50	79.51	2054	89.72
N=128 → k=32						
SPARE (full)	72.00	55.56	32.81	46.82	1763	88.11
w/o importance weighting	70.36	54.86	30.79	45.51	1650	87.62
w/o routing reconstruction ($\mathbf{M}$)	70.58	53.56	29.74	44.17	1608	86.93

D.5 GREEDY PIVOT DIAGNOSTICS

Algorithm 1 is a greedy approximation to the volume objective (4). On 48 synthetic weighted-Gram instances ($N \in [8, 14]$, $k \in [3, 5]$) where the optimum can be enumerated, it attains 93.6% of the optimal volume on average and recovers the exact maximum-volume subset in two thirds of the cases. For the Lebesgue constant $\Lambda = \|\mathbf{U}_k(\mathcal{I}, :)^{-1}\|_2$, which is the quantity that governs interpolation error, the greedy choice is within 1% of the best value in 70.8% of cases and within 7.9% on average.

The closed-form remapping satisfies its interpolation and subspace-consistency identities to machine precision on the same synthetic instances (maximum infinity-norm errors 1.9×10^{-15} and $1.7 \times$

Table 5: Left: accuracy when recompressed from COCO or TextVQA instead of GQA (Kimi-VL at k=48, Qwen3-VL at k=96). Right: effect of the number of calibration examples n at the tightest budget of each model. Avg. is aggregated as in Table 1, relative to the dense reference of the sweep of Table 4.

Calibration corpus			Calibration size n		
Model	Corpus	Avg.	Model	n	Avg.
Kimi-VL	GQA	97.84	Kimi-VL (k=32)	32	88.12
	COCO	98.04		128	87.68
	TextVQA	97.71		512	87.24
Qwen3-VL	GQA	98.38	Qwen3-VL (k=64)	32	92.85
	COCO	97.11		128	92.43
	TextVQA	98.79		512	92.86

Table 6: Text-only transfer on Qwen3-30B-A3B: the eight lm-eval tasks of the Sub-MoE protocol, compressed with the same code and calibration procedure as the VLM experiments (128 C4 documents). Avg. is the arithmetic mean over the eight tasks.

Model	ARC-C	ARC-E	BoolQ	HellaSwag	MMLU	OpenBookQA	RTE	Winogrande	Avg.
Dense	56.57	79.29	88.62	77.60	77.71	46.00	82.67	70.01	72.31
SPARE (k=96)	53.28	73.14	87.34	75.40	73.03	42.60	81.23	68.98	69.38
SPARE (k=64)	40.69	62.48	82.56	65.08	53.34	35.60	76.17	65.67	60.20
REAP (k=96)	54.01	74.83	87.77	75.38	72.33	43.80	79.06	69.61	69.60
REAP (k=64)	41.21	62.25	83.06	66.09	53.70	37.60	74.73	64.96	60.45
Sub-MoE (k=96)	29.35	48.82	71.10	49.34	47.39	31.00	63.54	56.35	49.61
Sub-MoE (k=64)	23.63	28.20	46.27	29.20	22.89	29.00	51.62	51.38	35.27
HC-SMoE (k=96)	29.69	48.82	70.86	49.39	47.97	32.00	64.26	55.72	49.84
HC-SMoE (k=64)	23.46	28.45	46.06	28.97	23.03	28.40	51.99	52.17	35.32

10^{-15}). The core condition numbers are 1.61, 2.48, 3.92, and 7.43 for $N = 16, 32, 64, 128$ at $k = N/2$.

D.6 REAL-MODEL ORACLE

We also test the pivot on real calibration geometry, using the Kimi-VL TextVQA calibration statistics. For each of the 26 MoE layers we enumerate all $\binom{12}{6} = 924$ subsets of a restricted candidate pool: the union of the six experts selected by SPARE and the six selected by Freq-Prune, filled to 12 with the most frequent remaining experts. Both methods are scored with the same weighted maximum-volume objective used by the pivoting rule. SPARE attains the restricted optimum in all 26 layers, whereas Freq-Prune attains it in none, with a mean gap of 19.84 log-volume units (Figure 4).

D.7 SELECTION AND REMAPPING SHARE THE CORE

The remapping $\mathbf{M} = \mathbf{U}_k\, \mathbf{U}_k(\mathcal{I}, :)^{-1}$ inverts the core selected in Algorithm 1, so its size is governed by the conditioning of that core: since $\|\mathbf{M}^\top \mathbf{g}\|_1 \le \|\mathbf{M}\|_\infty \|\mathbf{g}\|_1$, $\|\mathbf{M}\|_\infty$ bounds how much routing mass the remapped gate can create. The pivot keeps the core well conditioned; a core chosen by any other rule need not be. To make this concrete we perturb only the selection: a fraction r of the budget is reserved for the most frequently activated experts (text plus visual counts), and the pivot fills the remaining slots.

Table 7 reports the resulting cores, computed offline from the GQA calibration statistics with the same weighting and precision as the compression code. With r=0 the core is nearly orthogonal (median condition number 1.1–1.6) and $\|\mathbf{M}\|_\infty \approx$ 1–2: the routing mass of a discarded expert is redistributed with modest weights. Every reservation breaks this. At r=0.5 the median condition number is 123–332 and $\|\mathbf{M}\|_\infty$ is in the hundreds, with 13–26% of its entries negative. Frequently activated experts respond to the same common inputs, so their rows of $\mathbf{U}_k$ are nearly collinear, and $\mathbf{M}$ expresses a discarded expert as a large cancelling difference of near-parallel retained ones.

Such a remapping cannot be used at inference. The router keeps only the top-k_{top} entries of $\mathbf{M}^\top \mathbf{g}$, which truncates the cancellation; without renormalization the surviving weights are amplified by up to $\|\mathbf{M}\|_\infty$ in every layer and generation diverges, and with the gate clipped at zero and renormalized to its original mass the choice of experts is still dominated by the cancelling coefficients. Table 8

Table 7: Conditioning of the core $\mathbf{U}_k(\mathcal{I}, :)$ and size of $\mathbf{M}$ when a ratio r of the budget is reserved for the most frequently activated experts before pivoting; $r{=}0$ is SPARE. Medians over MoE layers, computed offline from the GQA calibration statistics.

	$\mathrm{cond}(\mathbf{U}_k(\mathcal{I}, :))$ / $\|\mathbf{M}\|_\infty$			
Model	$r{=}0$	$r{=}0.25$	$r{=}0.5$	$r{=}0.75$
Kimi-VL ($k{=}32$)	1.1 / 1.0	91 / 120	332 / 437	590 / 1090
Qwen3-VL ($k{=}64$)	1.6 / 2.1	72 / 101	180 / 340	379 / 757
Qwen3-VL ($k{=}96$)	1.4 / 1.8	53 / 83	123 / 213	241 / 537

Table 8: Qwen3-VL at $k{=}64$: the same selections routed with $\mathbf{M}$ or by plain restriction of the gate to the retained experts (w/o $\mathbf{M}$). The $r{=}0.5$ row with $\mathbf{M}$ uses the clipped and renormalized gate; without renormalization generation diverges. Avg. over the three tasks, relative to dense on the same tasks.

Selection	Routing	TextVQA	ChartQA	MMStar	Avg.(%)
Pivot ($r{=}0$, SPARE)	$\mathbf{M}$	81.07	70.40	63.06	91.99
Pivot ($r{=}0$)	w/o $\mathbf{M}$	80.91	69.28	60.50	90.34
Hard-keep $r{=}0.5$ + pivot	$\mathbf{M}$	0.26	0.00	4.14	1.89
Hard-keep $r{=}0.5$ + pivot	w/o $\mathbf{M}$	79.81	66.80	57.73	87.62

isolates the cause on Qwen3-VL at $k{=}64$: with $r{=}0.5$ and $\mathbf{M}$ the model collapses (MMStar 4.1, below chance), whereas the same retained experts routed by plain restriction of the gate recover to 87.62% of dense. The two effects are thus separable. Reserving frequent experts costs 2.7 points against a pure pivot selection under the same routing (87.62% vs. 90.34% without $\mathbf{M}$), consistent with Insight 1; inverting a core that the pivot did not choose turns that moderate loss into a collapse. On the pivoted core the same remapping is instead an asset, adding 1.7 points over plain restriction of the gate (91.99% vs. 90.34%): what decides between the two outcomes is the conditioning of the core, not the remapping itself. This is why SPARE derives the selection and the remapping from the same pivoted core rather than attaching $\mathbf{M}$ to an arbitrary selection rule.

D.8 QUALITATIVE TEXTVQA CASES

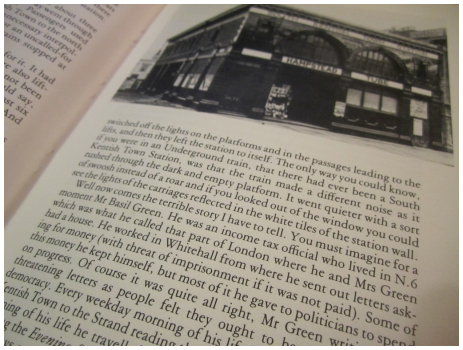
Figure 6 shows four TextVQA validation examples at 50% expert retention in which SPARE matches the annotated answer while both scalar pruning baselines do not. The examples span numerical reading, fine-grained OCR, and locating the relevant text in a cluttered scene. The baseline errors alter exact textual or numerical content (a–c) or fail to isolate the relevant label (d), illustrating the more graceful degradation of SPARE under aggressive compression seen in the aggregate scores.



(a) Calendar year



(b) Airline name



(c) First name in a passage



(d) Country on a bottle label

Case	Model	Question	GT	SPARE	REAP	Freq-Prune
(a)	Kimi-VL	what is the year on the calender?	2010	2010	2015	2016
(b)	Kimi-VL	what airline is the plane from?	lufthansa	lufthansa	luthansa	lufthans
(c)	Kimi-VL	what was mr. green's first name?	basil	basil	mr	mr green 's first name was mr green
(d)	Qwen3-VL	what country is listed?	india	india	no	theo country is listed is theo

Figure 6: TextVQA validation cases and model outputs at 50% expert retention. "GT" is the majority human answer. Kimi-VL retains k=32 of 64 experts; Qwen3-VL retains k=64 of 128.