YOU ONLY NEED 2/3 OF THE CHOSEN EXPERTS: AN EMPIRICAL STUDY OF DYNAMIC EXPERT PRUNING IN FINE-GRAINED MOE LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained mixture-of-experts (MoE) architectures have become a mainstream design for open-weight LLMs, with hundreds of experts and increasingly many selected per token. This shift makes dynamic expert pruning an attractive route to cheaper inference. Yet existing evidence comes largely from coarser architectures and likelihood-scored multiple-choice benchmarks, leaving three central questions open in the fine-grained regime: how redundant per-token expert selection is, how effectively existing pruning methods exploit that redundancy, and what governs a model's sensitivity to pruning. We fill this gap with a systematic empirical study of twelve fine-grained MoE checkpoints spanning nine architecture families, with a core suite of eleven benchmarks covering knowledge QA, mathematics, code generation, and general reasoning. We find that expert selection is far more redundant than the field's operating points assume: uniformly retaining about two thirds of the selected experts preserves 98.8% of unpruned performance on average, requiring only a one-integer change and delivering 1.2–$1.7\times$ measured speedup across two serving backends. This simple baseline leaves little room for dynamic allocation at conservative budgets: even the best published rules differ from it by under 1% at matched expert budgets. Their value emerges under aggressive pruning, where the best rules recover up to 3.0% over uniform truncation, with gains concentrated in the generative tasks that suffer the sharpest degradation. Sensitivity to aggressive pruning also depends on the model: larger and thinking models are more resilient, whereas multimodal models are more vulnerable. Together, these findings reveal how much expert computation fine-grained MoEs can dispense with, and establish when dynamic allocation earns its complexity, informing both practical deployment and future pruning methods.

1 INTRODUCTION

Fine-grained mixture-of-experts (MoE) architectures (Dai et al., 2024) have become a mainstream approach to efficient LLM scaling, combining large pools of small experts with sparse per-token activation. Qwen3-30B-A3B activates 8 of 128 routed experts, DeepSeek-V4-Pro 6 of 384, and Kimi K3 16 of 896. This design makes routing a richer allocation problem, with more choices over which combinations of experts serve each token. It also opens new opportunities for dynamic expert pruning, raising a basic question: **how much of the selected computation is actually necessary?**

A top-K router assigns the same expert count to every token, regardless of how concentrated its routing scores are. Dynamic expert pruning makes that count adaptive. Fine-grained architectures are especially suited to this approach: their longer lists of selected experts offer finer budget choices, while each skipped expert removes a feed-forward evaluation. The promise is to spend expert computation where it matters most. Establishing its value requires separating the redundancy in the native budget from the benefit of reallocating a reduced one. Uniform truncation provides the reference: what additional capability does dynamic allocation preserve at a comparable expert budget?

Evidence for dynamic expert pruning has not kept pace with the shift to fine-grained MoE. Most existing studies focus on coarser architectures and likelihood-scored multiple-choice benchmarks. Ranking candidate answers reveals little about whether the same pruning budget preserves a long

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

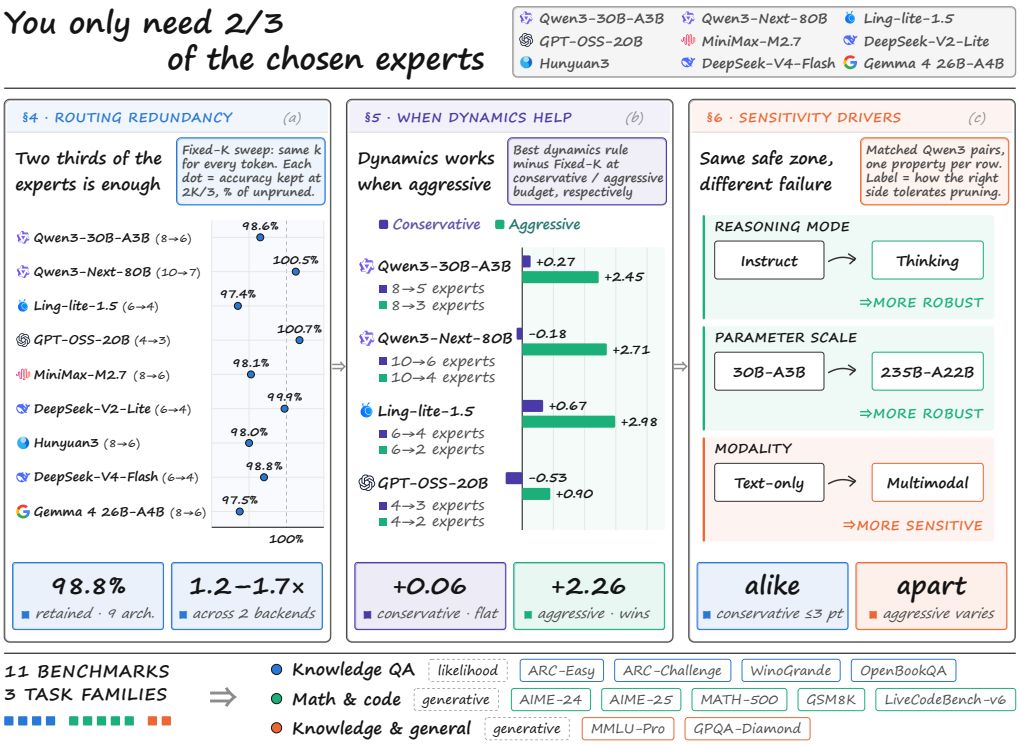

Figure 1: Overview. **(a)** Uniformly retaining approximately two thirds of the selected experts preserves 98.8% of unpruned performance on average. **(b)** The best dynamic rules add little over uniform truncation at conservative budgets but recover significantly under aggressive pruning. **(c)** Paired comparisons show that thinking and larger models are more robust, whereas the multimodal model is more sensitive.

mathematical derivation or a working program. Methods are also evaluated on different models and at different budgets, obscuring the added value of dynamic allocation. What the field lacks is a systematic account of both: how much expert selection can be removed, and when a dynamic rule is needed to remove it. We address this gap through three questions:

> **Q1.** How much redundancy does per-token expert selection carry in fine-grained MoE?
> **Q2.** When does dynamic allocation improve on simply lowering the expert count?
> **Q3.** What governs a model's sensitivity to expert pruning?

We study twelve checkpoints spanning nine architecture families. We first use uniform budget sweeps to characterise routing redundancy, then compare dynamic pruning rules against this baseline to measure the gains from adaptive allocation. Finally, paired model comparisons examine how reasoning mode, parameter scale and modality shape pruning sensitivity. The core evaluation covers four likelihood-scored QA benchmarks and seven generative benchmarks in mathematics, code and general reasoning. The study yields three central findings, summarised in Figure 1:

- **A systematic characterisation of routing redundancy.** We show that fine-grained expert selection is far more redundant than the field's operating points assume. Uniformly retaining approximately two thirds of the selected experts preserves 98.8% of unpruned performance on average across the nine models and three task families. This requires only lowering a single integer: no calibration, no token-specific allocation, and no change to the router's ranking. Much of the apparent opportunity for dynamic pruning is therefore already captured by the simplest baseline.
- **A reassessment of dynamic pruning across budgets.** We establish when dynamic allocation earns its complexity beyond simply lowering the expert count. At comparable average expert counts, even the best published rules remain within one percentage point of uniform truncation

under conservative pruning. Their advantage emerges under aggressive pruning, where the best rules recover up to 3.0 points and the winning rule varies across models and budgets. The opportunity for a pruning method therefore lies where uniform truncation begins to lose capability.

- **Sensitivity analysis and actionable guidance.** We identify how task and model properties shape pruning sensitivity. Generative tasks deteriorate more sharply than likelihood-scored QA and account for most of the gains from dynamic allocation. This sensitivity also varies across models, with differences emerging mainly under aggressive pruning: larger and thinking models are more robust, whereas multimodal models are more sensitive. These findings guide model-specific budget selection and highlight generative tasks as a key test of dynamic allocation.

These findings establish a practical starting point for fine-grained MoE inference. Finally, building on these, we provide concrete guidance for applying dynamic expert pruning to fine-grained MoEs and identify directions for future research.

2 Related work

2.1 Fine-grained MoE

MoE architectures (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022) have evolved from routing among a few wide experts to combining many narrower ones. Mixtral activates 2 of 8 experts per token, each matching the width of a dense feed-forward network (Jiang et al., 2024). DeepSeek-MoE introduced finer expert segmentation and shared experts, expanding the combinations available to each token (Dai et al., 2024). This design has since become mainstream in open-weight LLMs, including Qwen3, Qwen3-Next, MiniMax-M2 and DeepSeek-V4, listed in Table 1, as well as GLM-5 and Kimi K3 (GLM-5 Team, 2026; Kimi Team, 2026).

Longer selections offer finer budget choices than a top-2 router allows. These changes make fine-grained MoE a compelling setting for revisiting expert redundancy: both the available expert budgets and the signals used to allocate them have evolved substantially.

2.2 Dynamic expert pruning

Dynamic expert pruning retains the expert pool while reducing the subset executed at inference time, complementing structural pruning, which removes experts permanently (Muzio et al., 2024).

Existing rules differ in the information they use. NAEE skips the second expert when its gate weight falls below a calibrated fraction of the leading expert's (Lu et al., 2024). DiEP incorporates output similarity into this comparison, making retention depend on both router preference and overlap between expert computations (Bai et al., 2025b). Cumulative-probability routing, represented here by DynRoute, moves from pairwise comparisons to the selected distribution as a whole: it retains the shortest prefix whose cumulative gate weight reaches a target (Huang et al., 2024). BAN combines an offline layer-sensitivity profile with an online token signal, allocating the expert budget across tokens and depth with fine-grained routers and long-form reasoning in view (Chen et al., 2026). MoDES extends dynamic expert pruning to multimodal MoE, applying separate thresholds to text and vision tokens so allocation reflects modality differences (Huang et al., 2026).

Our study compares these allocation signals across fine-grained architectures and task families at comparable expert budgets to establish their value beyond uniform truncation.

3 Experimental setup

We use the routed expert budget to distinguish native routing redundancy from the benefit of dynamic allocation, and to compare pruning sensitivity across tasks and model properties.

3.1 Routing and expert budgets

For each token $\boldsymbol{x}$, an MoE router selects K of E routed experts, with indices e_i ordered by gate weight $w_1 \geq \cdots \geq w_K$. Pruning retains a subset of these ranks, $\mathcal{S}(\boldsymbol{x}, \ell)$, at layer ℓ. After renormal-

Table 1: Models. Total and active are parameter counts; E is the routed experts per layer, K how many the router natively activates per token, and n_s the shared experts every token passes through.

Model	Total	Active	E	K	n_s	Role
Core models uniform sweeps (§4) and dynamic rules (§5)						
Qwen3-30B-A3B-Instruct (Qwen Team, 2025a)	30.5B	3.3B	128	8	0	reference model
Qwen3-Next-80B-A3B-Instruct (Qwen Team, 2025b)	80B	3B	512	10	1	widest router
Ling-lite-1.5-2507 (Ling Team, 2025)	16.8B	2.75B	64	6	2	shared experts
GPT-OSS-20B (OpenAI, 2025)	21B	3.6B	32	4	0	narrowest router
Additional architectures uniform sweeps only (§4)						
MiniMax-M2.7 (MiniMax, 2026)	229B	11B	256	8	0	sigmoid router
DeepSeek-V2-Lite-Chat (DeepSeek-AI, 2024)	15.7B	2.4B	64	6	2	earliest fine-grained
Hy3 (Tencent Hy Team, 2026)	295B	21B	192	8	1	largest model
DeepSeek-V4-Flash-0731 (DeepSeek-AI, 2026)	165B	11.5B	256	6	1	sqrt-softplus router
Gemma 4 26B-A4B (Gemma Team, 2026)	25.2B	3.8B	128	8	1	GELU experts
Single-factor checkpoints one property changed (§6)						
Qwen3-30B-A3B-Thinking (Qwen Team, 2025a)	30.5B	3.3B	128	8	0	reasoning mode
Qwen3-235B-A22B-Instruct (Qwen Team, 2025a)	235B	22B	128	8	0	parameter scale
Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025a)	30B	3B	128	8	0	modality

ising the retained weights, the layer output is

$$\widetilde{\mathrm{MoE}}_\ell(\boldsymbol{x}) = \sum_{i \in \mathcal{S}(\boldsymbol{x},\ell)} \frac{w_i}{\sum_{j \in \mathcal{S}(\boldsymbol{x},\ell)} w_j} f_{e_i}(\boldsymbol{x}) + \sum_{j=1}^{n_s} f_j^{\mathrm{shared}}(\boldsymbol{x}). \tag{1}$$

Here, f_{e_i} is the expert at rank i, and n_s counts shared experts, which remain active and are excluded from the routed budget $k(\boldsymbol{x}, \ell) = |\mathcal{S}(\boldsymbol{x}, \ell)|$. FIXED-K retains the top k experts for every token and layer; dynamic rules choose the retained set using signals that vary across tokens and layers.

3.2 MODELS

Table 1 lists the twelve checkpoints and their roles. Nine models cover nine architecture families, with 32–512 routed experts and native selections of 4–10, including softmax, sigmoid and sqrt-softplus routers and models with and without shared experts. All nine support uniform sweeps (§4); four also support dynamic-rule comparisons (§5). Three additional Qwen3 checkpoints enable paired comparisons of reasoning-mode post-training, parameter scale and modality (§6). The four checkpoints in these pairs share $E = 128$ and $K = 8$, giving a common expert-budget axis.

3.3 TASKS AND EVALUATION

The core suite contains eleven benchmarks in three task families. Knowledge QA uses zero-shot likelihood scoring; the two reasoning families are evaluated through generated answers.

- **Knowledge QA:** ARC-Easy and ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2019), and OpenBookQA (Mihaylov et al., 2018) assess the model's command of scientific and commonsense knowledge.
- **Mathematical and code reasoning:** AIME-24 and AIME-25 (Math-AI, 2025), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), GSM8K (Cobbe et al., 2021), and LiveCodeBench-v6 (Jain et al., 2024) test multi-step reasoning.
- **Knowledge and general reasoning:** MMLU-Pro (Wang et al., 2024) and GPQA-Diamond (Rein et al., 2023) assess broad subject knowledge and its application to reasoning across domains.

The modality comparison additionally uses nine multimodal benchmarks and a text QA suite expanded to nine datasets, detailed in Appendix E.

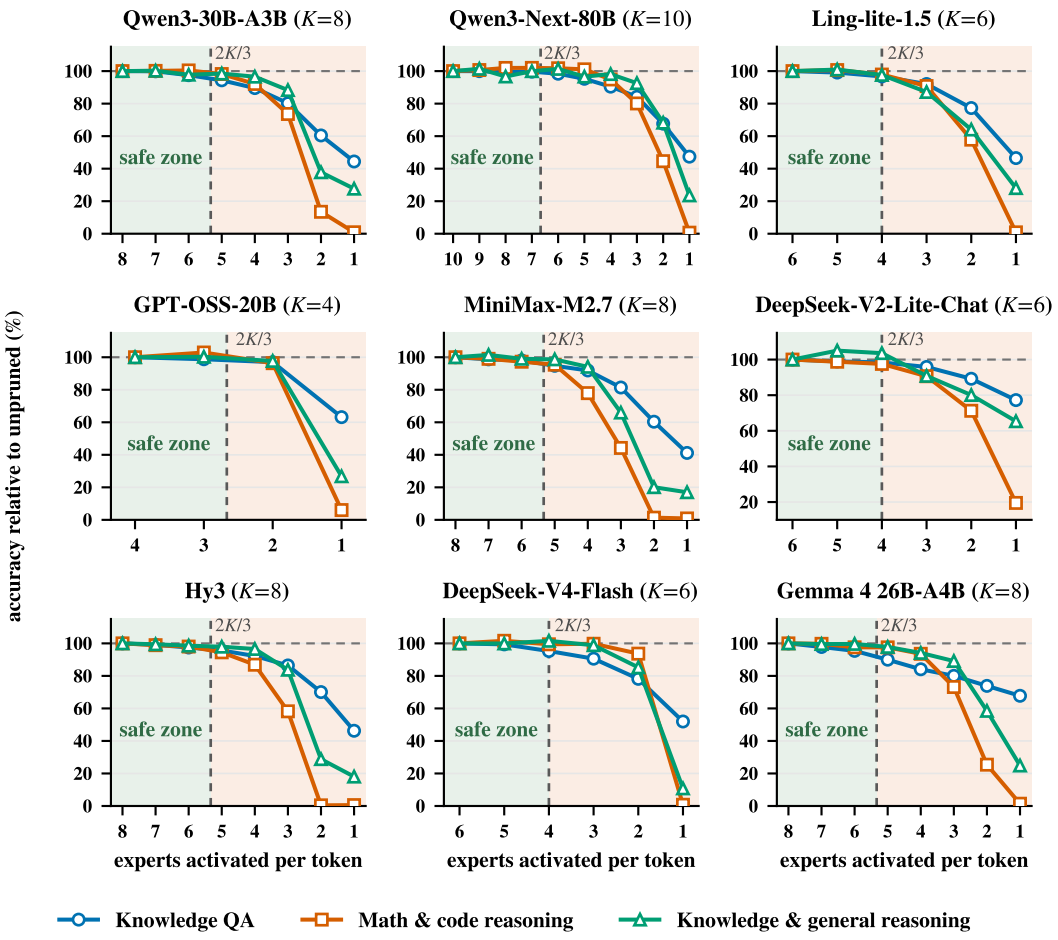

Figure 2: Expert-selection redundancy under uniform truncation. Each panel is one model, with its own native K on the horizontal axis and the budget tightening to the right; curves are the mean over the datasets of a task family, normalised to that family's score on the unpruned model.

4 How much routing does a fine-grained MoE actually need?

We lower the expert count uniformly across tokens and layers, keeping model parameters and the router's ranking unchanged. We sweep the retained count from the native selection down to one, measuring each task family's performance relative to its unpruned baseline. This one-integer intervention measures the redundancy in native routing and establishes the uniform baseline against which we evaluate dynamic allocation.

4.1 Two thirds of the selected experts are enough

Uniformly retaining $\lceil 2K/3 \rceil$ experts preserves 98.8% **of unpruned performance**, averaged over nine models and three task families (Figure 2). Six of the twenty-seven model-family combinations match or exceed their unpruned scores, including mathematical and code reasoning on both Qwen3 models and GPT-OSS-20B. Across architectures with native selections of four to ten experts per token, substantial savings are therefore available through a one-integer change. Put plainly: throw away roughly the lowest-weight third of the selected experts, and the model barely notices.

The gate-weighted mixture helps explain this tolerance (Equation 1). Fixed-K removes the lowest-weight choices and renormalises the retained weights, concentrating the mixture on the router's preferred experts. Removing a third of the selected experts therefore need not remove a third of their contribution.

Table 2: Measured speedup from uniform truncation at the tested reduced budgets. Throughput ratios are relative to the same unpruned model on each backend, reported separately for prefill and decode. All runs use batch size 4, 1024-token prompts and 256 generated tokens.

	Operating point			vLLM		HF Transformers	
Model	K	$\bar{k}$	ρ	prefill	decode	prefill	decode
Qwen3-30B-A3B	8	5	0.63	1.15	1.18	1.26	1.70
Qwen3-Next-80B	10	6	0.60	1.25	1.65	1.31	1.73
Ling-lite-1.5	6	4	0.67	1.21	1.38	1.21	1.52
GPT-OSS-20B	4	3	0.75	1.25	1.23	1.24	1.44

As the budget tightens, this spare capacity runs out. Qwen3-30B-A3B retains 73.6% of its mathematical and code reasoning score at three of eight experts, but falls to 13.5% after removing just one more. A modest further reduction can therefore exhaust the remaining tolerance. Yet the same absolute count has very different consequences across models: GPT-OSS-20B retains 96.3% with two experts. Two experts are a quarter of Qwen3's native selection and half of GPT-OSS's. The retained fraction of the native selection is therefore essential to interpreting a pruning budget.

4.2 Knowledge survives what reasoning does not

Budget tightening also changes which capabilities are most resilient. Knowledge QA declines gradually, while the generative families initially hold steady and then fall sharply. On Qwen3-30B-A3B, knowledge QA retains 97.5% of its unpruned score at six experts, below both generative families. At two experts, that ordering reverses: knowledge QA retains 60.4%, while mathematical and code reasoning falls to 13.5%. On eight of the nine models, mathematical and code reasoning is the least robust family at the aggressive end.

This reversal separates retaining knowledge from sustaining a complete solution. A derivation or program must carry that knowledge through a sequence of dependent steps. Each generated step becomes context for the next, allowing local disruptions to propagate through the answer. A model can therefore retain much of what it knows while losing the ability to use it in an extended solution.

4.3 From routing redundancy to inference speed

The redundancy translates into measurable inference gains. Across the four core models at the tested reduced budgets in Table 2, FIXED-K improves decode throughput by 1.18–$1.65\times$ on vLLM and 1.44–$1.73\times$ on HF Transformers. Prefill throughput also improves on both backends.

The two backends expose different costs for executing the retained token–expert assignments. In our HF Transformers implementation, experts run through separate, unfused GEMMs, while vLLM's fused MoE kernels already amortise part of the execution overhead. FIXED-K reduces routed expert computation and its associated data movement in both cases. Even with fused execution, Qwen3-Next-80B gains 65% in decode throughput. The magnitude varies across models and backends, but the practical result is clear: substantial routing redundancy can be converted into faster inference by changing a single integer.

> **Takeaway.** *Uniformly retaining roughly two thirds of the selected experts preserves 98.8% of unpruned performance on average. Its limits emerge under aggressive pruning, where long-form reasoning deteriorates sharply. Dynamic allocation now faces its decisive test: can redistributing the same budget preserve the reasoning uniform truncation loses?*

5 When does dynamic allocation beat uniform pruning?

Uniform pruning establishes how much expert computation can be removed with a single budget choice. We now ask when allocating that budget dynamically preserves more capability. We compare four published rules with FIXED-K on four fine-grained MoEs, first across budget tiers and then across task families.

Table 3: Iso-budget comparison, per dataset. Each model is given two budgets, and under each the uniform baseline is paired with the best of the four dynamic rules at that same budget, selected by the mean over all eleven datasets (last column). The superscript is that rule's margin over the FIXED-K row directly above it. The budget beside a rule is its measured $\bar{k}$ for generative / QA, since the knob is solved separately per regime (§5.1); rules spending more than 10% over the tier are excluded.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
Qwen3-30B-A3B, natively $K{=}8$												
unpruned, $k{=}8$	60.8	85.0	73.2	32.0	89.6	74.1	61.4	94.4	41.1	74.2	56.6	67.49
Conservative budget, $\bar{k}\approx 5$ ($\rho{=}0.62$)												
FIXED-K, $k{=}5$	54.9	81.4	69.5	31.0	91.0	72.1	55.4	93.9	41.7	73.8	55.0	65.43
DYNROUTE, $4.91/5.09$	56.5	81.4	71.6	31.0	89.6	73.2	55.2	94.2	41.7	73.2	55.1	**65.70**$^{+0.27}$
Aggressive budget, $\bar{k}\approx 3$ ($\rho{=}0.38$)												
FIXED-K, $k{=}3$	44.6	71.8	58.0	27.0	80.2	40.0	31.1	88.5	25.7	67.0	48.5	52.95
BAN, $3.14/3.08$	46.1	72.8	61.5	25.8	88.0	42.9	34.0	90.4	28.6	68.8	50.5	**55.40**$^{+2.45}$
Qwen3-Next-80B, natively $K{=}10$												
unpruned, $k{=}10$	63.1	86.9	76.2	34.2	88.4	79.7	67.1	95.2	53.1	82.1	74.2	72.75
Conservative budget, $\bar{k}\approx 6$ ($\rho{=}0.60$)												
FIXED-K, $k{=}6$	63.1	85.6	73.9	33.6	90.8	82.2	66.6	95.7	54.9	82.1	76.3	73.16
BAN, $6.18/6.08$	62.4	85.7	74.1	32.4	89.2	82.7	69.7	95.3	54.9	82.2	74.2	$72.98^{-0.18}$
Aggressive budget, $\bar{k}\approx 4$ ($\rho{=}0.40$)												
FIXED-K, $k{=}4$	55.9	82.2	67.2	30.2	88.0	74.8	58.8	93.1	49.1	80.7	72.7	68.43
BAN, $4.27/4.18$	59.8	83.5	67.8	32.8	88.6	79.4	64.6	94.4	57.1	81.3	73.2	**71.14**$^{+2.71}$
Ling-lite-1.5, natively $K{=}6$												
unpruned, $k{=}6$	60.2	82.3	70.3	31.6	89.2	40.9	28.1	91.4	34.3	69.4	56.6	59.48
Conservative budget, $\bar{k}\approx 4$ ($\rho{=}0.67$)												
FIXED-K, $k{=}4$	58.3	81.2	66.1	30.8	89.0	38.7	26.0	90.5	33.7	67.4	55.6	57.94
DYNROUTE, $4.17/4.22$	58.4	82.0	68.4	30.4	90.2	40.1	26.8	90.5	32.0	67.3	58.6	**58.61**$^{+0.67}$
Aggressive budget, $\bar{k}\approx 2$ ($\rho{=}0.33$)												
FIXED-K, $k{=}2$	38.9	69.6	57.4	23.0	70.4	5.0	8.4	72.4	8.0	42.3	38.4	39.44
BAN, 2.05	42.2	71.2	57.1	24.6	70.4	10.2	11.5	75.1	14.9	47.5	41.9	**42.42**$^{+2.98}$
GPT-OSS-20B, natively $K{=}4$												
unpruned, $k{=}4$	45.3	77.5	66.9	27.2	90.6	73.7	71.7	86.2	58.3	74.3	64.6	66.94
Conservative budget, $\bar{k}\approx 3$ ($\rho{=}0.75$)												
FIXED-K, $k{=}3$	44.8	76.7	65.4	27.4	91.0	77.4	74.5	87.0	61.7	73.6	65.7	67.74
DYNROUTE, 2.97	43.4	77.8	65.1	28.2	90.0	77.0	72.4	86.6	62.3	73.5	63.1	$67.21^{-0.53}$
Aggressive budget, $\bar{k}\approx 2$ ($\rho{=}0.50$)												
FIXED-K, $k{=}2$	44.3	75.9	64.1	26.0	86.6	73.9	69.4	88.7	48.0	70.5	65.2	64.79
DYNROUTE, 2.07	42.3	73.6	63.1	26.1	89.9	73.7	71.2	85.6	62.5	70.8	63.6	**65.69**$^{+0.90}$

5.1 COMPARING ALLOCATION AT MATCHED BUDGETS

We implement NAEE, DYNROUTE, DIEP and BAN in a common router patch on vLLM. For DIEP, we generalise its similarity-adjusted skipping criterion to fine-grained routing and also test a variant that damps the similarity term. Appendices A and B detail the rules and calibration.

Each configuration is characterised by $\bar{k}$, the average number of active routed experts measured inside the router during evaluation, and by the relative budget $\rho = \bar{k}/K$ (Appendix A.4). Thresholds are calibrated separately for generation and likelihood-scored QA, where the same setting can spend different budgets. At each tier, Table 3 pairs FIXED-K with the eligible rule scoring highest over all eleven datasets. Configurations exceeding the target budget by more than 10% are excluded.

5.2 DYNAMIC ALLOCATION PAYS OFF UNDER AGGRESSIVE PRUNING

At conservative budgets, even the best dynamic rule offers little advantage. Its margin over FIXED-K ranges from -0.53 to $+0.67$ points across the four models, averaging just $+0.06$. The redundancy identified in §4 leaves little for adaptation to recover: a uniformly reduced expert count already captures most of the available benefit.

Tightening the budget changes the result on every model. At target budgets retaining one third to one half of the native selection, the best rule improves on FIXED-K by up to 3.0 points, averaging $+2.26$. Allocation now recovers accuracy that uniform truncation loses at the corresponding budget.

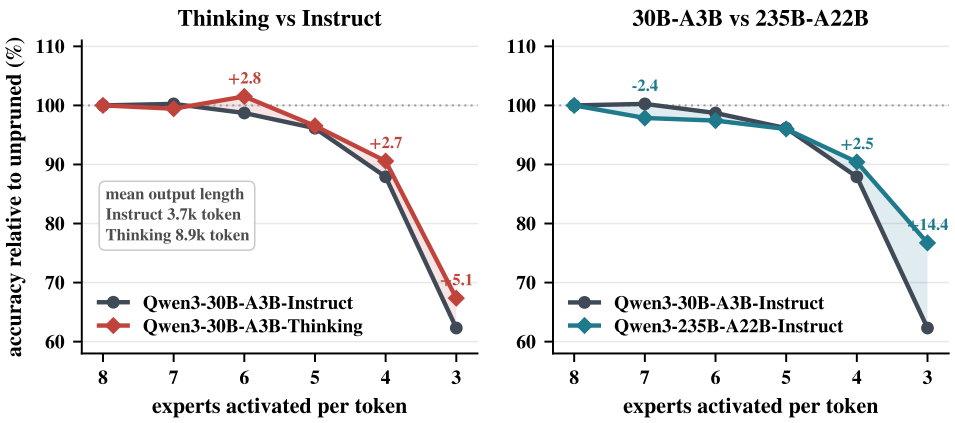

Figure 3: Two matched pairs under uniform truncation, each curve normalised to that model's own unpruned score. Curves are the mean over the four hardest generative datasets, AIME-24, AIME-25, LiveCodeBench and GPQA-Diamond. The axis stops at three experts, below which both members of each pair have collapsed on these tasks.

This is where dynamic pruning earns its place: once the expert budget is tight enough to damage performance, deciding where to spend it becomes consequential.

The budget also changes which rule performs best. On Qwen3-30B-A3B, DynRoute leads at the conservative tier, whereas Ban wins at the aggressive tier. Ban wins on three of the four models under aggressive pruning, highlighting calibrated layer sensitivity as a promising allocation signal. An evaluation at a single operating point can therefore miss both the gains from adaptation and the rule that delivers them.

5.3 Generative tasks reveal the value of reallocation

These aggregate gains are concentrated on generative tasks. At Qwen3-30B-A3B's aggressive budget, Ban improves the mean over seven generative datasets by 3.17 points, compared with 1.20 on the four Knowledge QA datasets. On GPT-OSS-20B, DynRoute raises LiveCodeBench from 48.0 to 62.5, a 14.5-point gain within a suite-wide improvement of 0.90 points.

These results connect the gains from allocation to the task split in §4. Generation exposes a sharper loss under uniform pruning and a larger recovery when the remaining budget is distributed dynamically. Knowledge QA alone captures only part of that change. An evaluation dominated by likelihood-scored answers can therefore understate both the cost of reducing the expert count and the benefit of adapting it. Requiring a complete derivation, program or answer reveals where the allocation decision has the greatest effect.

> **Takeaway.** *Dynamic allocation adds little at conservative budgets but recovers capability under aggressive pruning, especially on generative tasks. Budget determines when adaptation pays off, while the evaluation tasks determine how much of that value is visible.*

6 What shapes sensitivity to expert pruning?

The budget dependence in §4 and §5 raises a further question: which model properties allow pruning to go further? We examine reasoning mode, parameter scale and modality through three pairs drawn from four Qwen3 checkpoints, all with 128 routed experts and a native selection of eight. Figure 3 presents the reasoning-mode and scale comparisons on four challenging generative datasets, with each curve normalised to its own unpruned performance. The modality comparison and its accompanying routing analysis are reported in Appendix G.

6.1 THINKING PRESERVES MORE GENERATIVE CAPABILITY

Reasoning-mode post-training preserves more generative capability as the expert budget tightens. Qwen3-30B-A3B-Thinking and its Instruct counterpart retain similar fractions of their unpruned performance from eight to five experts. At four experts, the thinking model leads by 2.7 percentage points; at three, its advantage grows to 5.1 points.

This advantage accompanies longer generation: the thinking model averages 8.9k output tokens, compared with Instruct's 3.7k. Appendix E shows that the retention gain is concentrated in generative tasks, with little difference on Knowledge QA. Thinking therefore preserves more of the capabilities most vulnerable to pruning in §4, even while producing substantially longer solutions.

6.2 SCALE MATTERS MOST UNDER AGGRESSIVE PRUNING

Parameter scale produces an even larger separation as pruning deepens. On the same tasks, Qwen3-235B-A22B-Instruct holds no advantage over Qwen3-30B-A3B-Instruct from seven to five experts. At four experts, the larger model leads by 2.5 points. At three, it retains 76.7% of its unpruned performance against 62.3% for the smaller model, a 14.4-point gap.

Like the thinking advantage, this separation is concentrated in generative tasks, with no gain on Knowledge QA (Appendix E). Conservative pruning therefore conceals much of the difference: scale matters most when the reduced expert budget begins to constrain generative capability.

> **Takeaway.** *Model properties matter most under aggressive pruning. Reasoning-mode post-training and parameter scale preserve more generative capability, while the multimodal checkpoint degrades more sharply (Appendix G). These differences shape how far the expert budget can be reduced.*

7 IMPLICATIONS FOR DYNAMIC EXPERT PRUNING

Building on these findings, we provide practical guidance for selecting expert budgets and applying dynamic expert pruning, and identify directions for future methods that improve capability retention and inference efficiency.

Start with uniform truncation at $\lceil 2K/3 \rceil$ experts and lower the count step by step on the target model and workload (§4). For reasoning applications, score complete derivations, programs or answers at each step, so the sweep shows whether the model can still finish the task rather than whether it still recognises a correct answer. The resulting curve marks the range of budgets that meet the application's quality requirements, and the point at which capability starts to fall away. If the required savings lie beyond that range, dynamic allocation is what recovers capability at the tighter budget. BAN and DYNROUTE are the candidates to try first: calibrate their thresholds for the target workload, then compare them against FIXED-K at a matched average expert count.

The same comparison shows where new methods have room to improve: recovering the generative capability that uniform truncation loses at tight budgets, with layer sensitivity a promising complement to token-level routing signals. Uneven allocation also asks more of the serving stack, since kernels must execute a varying number of experts per token without spending the savings on overhead. Reporting retained capability together with end-to-end latency or throughput keeps both sides visible, showing what a rule is worth against uniform truncation at its real serving cost.

8 CONCLUSION

This study bridges the gap between the widespread adoption of fine-grained MoEs and the evidence available for dynamic expert pruning. We provide a systematic account of routing redundancy, the added value of dynamic allocation and model sensitivity across both likelihood-scored QA and generative tasks. This evidence supports practical pruning decisions and provides a common basis for assessing what future methods add beyond uniform truncation.

AI USE STATEMENT

We used a large language model to polish the writing of this paper, limited to improving grammar, phrasing and clarity in text written by the authors. It was not used for research ideation, experimental design, analysis or the derivation of any result reported here. The authors reviewed all edited text and take full responsibility for the content of the paper.

ETHICS STATEMENT

This work evaluates published expert pruning methods on publicly released open-weight models and public benchmarks. It involves no human subjects and no new data collection. Our aim is to advance the efficiency of MoE inference and thereby lower the cost of serving these models. We see no evident ethical risks arising from the study beyond those already associated with the open-weight models it evaluates.

REPRODUCIBILITY STATEMENT

Appendix A describes the router patch through which every rule is implemented, the evaluation harnesses used for scoring, and the offline calibration that two of the rules require. Appendix A.4 gives the procedure for solving each rule's threshold to a target budget and the check comparing achieved against predicted expert use. Appendices C to E report the measured $\bar{k}$ and the per-task scores behind every figure and table in the main text, and Appendix F the protocol behind the throughput measurements. The code is provided in the supplementary material and will be publicly released.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. DiEP: Adaptive mixture-of-experts compression through differentiable expert pruning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 56090–56115. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1878.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019.

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*, 2024.

Yuanteng Chen, Peisong Wang, Yuantian Shao, Nanxin Zeng, Chang Xu, and Jian Cheng. Ban&Pick: Enhancing performance and efficiency of MoE-LLMs via smarter routing. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026. URL https://arxiv.org/abs/2509.06346. To appear. arXiv:2509.06346.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://www.jmlr.org/papers/v23/21-0998.html.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness. https://doi.org/10.5281/zenodo.12608602, July 2024.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. *arXiv preprint arXiv:1802.08218*, 2018.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. LightEval: A lightweight framework for LLM evaluation. https://github.com/huggingface/lighteval, 2023.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.696.

Yushi Huang, Zining Wang, Zhihang Yuan, Yifu Ding, Ruihao Gong, Jinyang Guo, Xianglong Liu, and Jun Zhang. MoDES: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2511.15690.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *arXiv preprint arXiv:1902.09506*, 2019.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*, 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. In *International Conference on Learning Representations*, 2021.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Ling Team. Ling-lite-1.5-2507. `https://huggingface.co/inclusionAI/Ling-lite-1.5-2507`, 2025.

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*, 2023.

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 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6159–6172, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.334.

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. *arXiv preprint arXiv:2203.10244*, 2022.

Math-AI. AIME 2025. `https://huggingface.co/datasets/math-ai/aime25`, 2025.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. InfographicVQA. *arXiv preprint arXiv:2104.12756*, 2021.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

MiniMax. The MiniMax-M2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026.

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

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Qwen Team. Qwen3-Next-80B-A3B-Instruct. `https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Instruct`, 2025b.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. SocialIQA: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

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. *arXiv preprint arXiv:1904.08920*, 2019.

Tencent Hy Team. Hy3. `https://huggingface.co/tencent/Hy3`, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. *arXiv preprint arXiv:2311.16502*, 2023.

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

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. *arXiv preprint arXiv:2407.12772*, 2024a.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024b.

A Implementation

§5 compares four published dynamic expert pruning rules. We reimplemented all four on vLLM (Kwon et al., 2023) and extended them to the fine-grained routers of Table 1, and this appendix documents that work: §A.1 where the rules are applied, §A.2 what each of them computes, §A.3 the offline statistics two of them need, and §A.4 how a knob is turned into the target budget that makes the comparison of §5 an iso-budget one.

A.1 Where the rules are applied

All four rules are implemented against vLLM. This is a requirement rather than a preference: over half of our suite is long-form generation, and much of it at budgets where a pruned model rambles towards the token limit, so the sweeps reported here are only affordable on a serving stack with paged attention and fused expert kernels. It also means the rules are measured in the setting they would actually be deployed in, which is what makes the throughput discussion of §4.3 meaningful.

The intervention itself is small and sits in one place, at the point where the router's scores have been computed and the top K experts are about to be handed to the expert kernels. Each rule receives

Table 4: The rules compared in §5, all re-implemented in the one router patch of §A.1. Every rule keeps all E experts resident and changes only how many of the K selected ones are evaluated per token. "Layer budget" is whether the rule can spend differently at different depths; "calib." whether it needs the offline statistics of §A.3; the last two columns give whether the rule imposes a floor on the number of experts kept and the single knob the budget solver of §A.4 moves. Where a floor is present its value is set per configuration (§A.2) rather than fixed per rule.

Rule	Retention decision conditioned on	layer budget	needs calib.	expert floor	knob
FIXED-K (baseline)	a constant k for all tokens	✗	✗	✗	k
NAEE	gate weight relative to the top-1 expert	✗	✗	✓	β
DYNROUTE	cumulative retained probability mass	✗	✗	✗	p
DIEP	top-1 ratio scaled by expert similarity	✗	✓	✓	β
BAN	layer sensitivity + token sensitivity	✓	✓	✓	λ

that slice, in descending weight order, and returns how far down it to read; the retained gate weights are then renormalised exactly as in Equation 1. Because the hook is at the selection step rather than inside any model definition, one implementation covers all four architectures. Shared experts are outside it by construction: every token passes through them, no rule can drop them, and they are excluded from the budgets we report, so a $\bar{k}$ in this paper is always a count of routed experts. Ling-lite-1.5 has two of them, so a $\bar{k}$ of 2 there costs four expert evaluations and is not comparable with a $\bar{k}$ of 2 on a model without shared experts. Scoring runs on the standard harnesses throughout: Habib et al. (2023) for the generative benchmarks, Gao et al. (2024) for likelihood-scored QA and Zhang et al. (2024a) for the multimodal suite.

A.2 THE FOUR RULES AS IMPLEMENTED

Throughout, $w_1 \geq w_2 \geq \cdots \geq w_K$ are the softmax gate weights of the natively selected experts $e_1, \ldots, e_K$ at layer ℓ for one token, and $k_{\min}$ is a floor on the number of experts kept. It is part of the published form of NAEE, DIEP and BAN, and absent from DYNROUTE. Where it is present we do not hold it at one value per rule. A rule can never spend fewer than $k_{\min}$ experts on a token, so the floor bounds the average budget the rule can reach at all, and on a narrow router or at the tighter of the two tiers it has to be lowered before the target $\bar{k}$ becomes attainable. We therefore adjust it per budget, alongside the knob, as part of the solve of §A.4.

NAEE. A flat threshold relative to the leading expert, applied as a prefix. Rank i is kept if it is among the first $k_{\min}$, or if no rank after them up to and including i has fallen below the threshold:

$$\text{keep } i \iff i \leq k_{\min} \quad \text{or} \quad w_j \geq \beta\, w_1 \ \text{ for all } j \text{ with } k_{\min} < j \leq i. \tag{2}$$

Equivalently, reading down the ranking, retention stops at the first expert past the floor whose weight is less than a β fraction of the top-1 weight. The rule has one knob, β, and needs no calibration.

DYNROUTE. A threshold on cumulative retained mass rather than on any single pair of experts. The kept set is the shortest prefix of the ranking whose weights reach a target p:

$$k(\boldsymbol{x}, \ell) \;=\; \min\Big\{\, i \;:\; \textstyle\sum_{j \leq i} \hat{w}_j \geq p \,\Big\}, \tag{3}$$

falling back to K if the target is never reached. Two readings of $\hat{w}$ are implemented: the softmax weights as they are, which is the published form, and the same weights renormalised over the K retained ranks first, which makes p a fraction of the mass the router actually routes rather than of the whole distribution. The configurations reported in Table 3 use the renormalised reading, since it is the one whose p keeps its meaning across models with different K. The rule has one knob and needs no calibration.

DIEP. NAEE's threshold, scaled per expert by how similar that expert is to the leading one. With $\mathrm{sim}_\ell(\cdot,\cdot)$ a calibrated similarity between expert outputs at layer ℓ and $\overline{\mathrm{sim}}_\ell$ its mean over expert pairs at that layer,

$$\gamma_i \;=\; \Big(\frac{\mathrm{sim}_\ell(e_1, e_i)}{\overline{\mathrm{sim}}_\ell}\Big)^{\alpha}, \qquad \gamma_1 \equiv 1, \qquad \text{keep } i \iff i \leq k_{\min} \ \text{ or } \ w_i \geq \beta\, \gamma_i\, w_1. \tag{4}$$

An expert that duplicates the leading one faces a higher bar and is dropped sooner; one that contributes something different is kept on a smaller weight. The exponent α is ours, not the published rule's, and interpolates: $\alpha{=}1$ is the rule as published, $\alpha{=}0$ removes the similarity term and reduces Equation 4 exactly to Equation 2, and intermediate values damp the spread of γ without removing it. Rows marked DIEP-d in Appendix D use $\alpha < 1$. Whether the threshold is applied per rank independently or as a prefix is also a choice; we apply it independently, for the reasons given in Appendix B.

One limitation of our DIEP rows has to be stated plainly, because it changes how they should be read. The published method pairs this skipping criterion with a differentiable pruning objective; we implement only the criterion. Equation 4 is therefore our generalisation of one component of DIEP to a wide router, and the DIEP rows in Appendix D are evidence about that component under that generalisation, not about the quality DIEP as published can reach. This is also why we do not present those rows as a ranking of the method against the others in the main text.

BAN. The only rule here that varies the budget with depth as well as with the token. It converts two sensitivity signals into an integer expert count. The token signal is how much of the routed mass the leading three experts hold,

$$r \;=\; \frac{\sum_{i\le 3} w_i}{\sum_{i\le K} w_i}, \qquad s_{\mathrm{tok}} \;=\; \mathrm{clamp}\Big(\frac{r_{\max}-r}{r_{\max}-r_{\min}},\,0,\,1\Big), \tag{5}$$

so a token whose distribution is peaked scores low and a flat one scores high. This is combined with a calibrated per-layer sensitivity s_ℓ and mapped onto the available range of expert counts:

$$s \;=\; \mathrm{clamp}\Big(\tfrac{\lambda}{2}\big(s_\ell + s_{\mathrm{tok}}\big),\,0,\,1\Big), \quad k(\boldsymbol{x},\ell) \;=\; \mathrm{clamp}\Big(\mathrm{round}\big(k_{\min} + (K-k_{\min})\,s\big),\,k_{\min},\,K\Big). \tag{6}$$

The single knob λ scales the combined sensitivity and therefore the average budget. Note the direction of Equation 5: it is the tokens whose routers are least decided that receive the most experts, which is the mechanism §5.2 appeals to, and $r_{\min}$, $r_{\max}$ and s_ℓ all come from calibration.

A.3 OFFLINE CALIBRATION

DIEP and BAN need statistics that cannot be read off a single token's router output. Both are collected once per checkpoint, before any evaluation, on 512 documents of C4 under HuggingFace Transformers, and stored as an artifact that the serving-time patch loads. No evaluation data is used for calibration, and no evaluation run searches over hyperparameters.

For DIEP the artifact holds, per MoE layer, the pairwise linear CKA between expert outputs on the calibration tokens, which is the sim_ℓ of Equation 4, together with its mean over pairs. It also holds a per-layer median of w_2/w_1, which the published rule uses as a second scaling factor; we fold that level into β instead, since β is what the budget solver of §A.4 moves (see Appendix B).

For BAN the artifact holds the per-layer sensitivity s_ℓ and the range $[r_{\min}, r_{\max}]$. The former is obtained by comparing, at each layer in turn, the layer output under full top-K routing against its output when only the leading three experts are kept, measuring the divergence between the resulting next-token distributions, and normalising across layers; the latter is the observed minimum and maximum of r over the calibration tokens.

A.4 SOLVING EACH KNOB TO A TARGET BUDGET

A configuration is characterised by the average number of active routed experts it spends, measured inside the router over the evaluated tokens $\mathcal{T}$ and the L MoE layers of the model:

$$\bar{k} \;=\; \frac{1}{|\mathcal{T}|\,L}\sum_{\boldsymbol{x}\in\mathcal{T}}\sum_{\ell=1}^{L} k(\boldsymbol{x},\ell), \qquad \rho \;=\; \frac{\bar{k}}{K}. \tag{7}$$

Because a rule is compared by this budget rather than by its knob (§5.1), each knob has to be set to hit a target $\bar{k}$ before the run, not after. Doing this by trial evaluation would be prohibitive: a single configuration of the largest model is a multi-hour run. We therefore solve each knob offline against measured routing statistics.

For NAEE and DYNROUTE the map from knob to average expert count is estimated directly, by replaying the retention criterion over router outputs recorded on the calibration corpus across a grid of knob values, and the value whose predicted average is closest to the target is used. For DIEP the same grid is not available, because the threshold depends on γ; instead the weight distribution is summarised by fitting a Beta survival function to the NAEE sweep, the average under Equation 4 is predicted by integrating that survival function against the calibrated γ distribution, and β is found by binary search. For BAN the token sensitivity of Equation 5 is fitted with a Beta distribution and the rounding in Equation 6 is inverted numerically for λ.

Because a threshold is not a budget (§5.1), every knob is solved twice, once against the generative prompt distribution and once against the likelihood-scored one. This is why a rule in Table 3 and in Appendix D carries two measured budgets rather than one. The residual between the solved target and the realised $\bar{k}$ is what the budget columns of those tables report, and it is the reason rules that overspend their tier by more than 10% are excluded from the main-text comparison rather than quietly compared at a cheaper budget.

B GENERALISING A TOP-2 SKIPPING RULE TO TOP-K ROUTING

DIEP was published for a router that selects two experts per token, and the shape of the rule reflects that. With $K{=}2$ there is exactly one decision to make, whether to keep e_2, and exactly one similarity to consult, that between e_1 and e_2. Both the threshold and the calibrated statistic are therefore defined on the rank-2 expert specifically. Applying the rule to a router that selects eight or ten experts requires deciding what the rank-i analogue is, and the choice is not forced by the original.

Our generalisation is the one in Equation 4: every rank $i > 1$ is compared against the leading expert with its own similarity factor γ_i, and each rank is judged independently. The alternative applies the same per-rank threshold but stops at the first failure, as NAEE does, which keeps the retained set a prefix of the ranking. The two differ whenever a low-similarity expert sits below a high-similarity one: the independent form can keep rank 5 after dropping rank 3, the prefix form cannot. We use the independent form throughout because it is the more faithful reading of a rule whose whole content is a per-expert comparison.

The rule's own calibrated level is also rank-2-specific. The published form multiplies the threshold by a per-layer median of w_2/w_1, a quantity with no rank-i analogue that is not simply the distribution of w_i/w_1, which changes shape with K. We therefore absorb the level into β and let the budget solver of §A.4 set it, which is also what makes the comparison at matched $\bar{k}$ possible at all. Keeping the factor explicit and setting $\beta{=}1$ would stay closer to the original expression, but it pins the operating point to whatever level the calibration statistic implies and leaves no free parameter to solve, so such a configuration could not be placed at a matched budget alongside the other rules.

Two observations from the results, since they bear on the reading of Appendix D rather than on our implementation choices. The similarity term's contribution is not small: on Qwen3-30B-A3B at the conservative budget, damping it with $\alpha < 1$ moves the suite mean from 54.32 to 61.60, and removing it entirely, which is NAEE, gives 63.92. And the size of the deficit does not track router width in the way one might expect. Against FIXED-K at the conservative budget it is -11.27 on GPT-OSS-20B at $K{=}4$ and -11.11 on Qwen3-30B-A3B at $K{=}8$, but only -2.38 on Ling-lite-1.5 at $K{=}6$ and -2.16 on Qwen3-Next-80B at $K{=}10$, the widest router in the set. Whatever our generalisation costs, it is not a simple function of how many experts there are to generalise over, and we do not have a mechanism for the pattern.

C FULL RESULTS: UNIFORM TRUNCATION

Tables 5 and 6 give every FIXED-K budget on every dataset for all nine models, which is the data behind Figure 2 and behind the task-family curves discussed in §4. Reading down a block shows the two features the main text draws on: the near-flat region at the top, and the fact that the collapse arrives at a fraction of the native budget rather than at a fixed number of experts, so that $k{=}2$ is catastrophic on the $K{=}8$ model and almost free on the $K{=}4$ one.

Table 5: Uniform truncation, every budget and every dataset, on the four core models. Each row is one value of k applied to all tokens and all layers; the top row of each block is the unpruned model. Scores are each dataset's primary metric as a percentage and the last column is the unweighted mean over the datasets that were run: a dash marks a cell that was not, and it is left out of that row's mean. Green rows are above half the native K, where uniform truncation needs no rule to help it; orange rows are at or below it, past the crossover of §5. This table and Table 6 are what Figure 2 is drawn from.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
Qwen3-30B-A3B-Instruct, natively $K{=}8$												
unpruned, $k{=}8$	60.8	85.0	73.2	32.0	89.6	74.1	61.4	94.4	41.1	74.2	56.6	67.49
FIXED-K, $k{=}7$	60.3	84.0	73.2	33.2	89.2	74.6	61.3	94.7	41.3	74.5	56.6	67.54
FIXED-K, $k{=}6$	59.2	82.8	71.3	31.4	90.4	72.8	59.3	95.0	44.6	74.4	53.5	66.79
FIXED-K, $k{=}5$	54.9	81.4	69.5	31.0	91.0	72.1	55.4	93.9	41.7	73.8	55.0	65.43
FIXED-K, $k{=}4$	51.3	77.2	65.1	31.2	87.8	62.5	47.9	93.3	40.6	72.3	54.0	62.11
FIXED-K, $k{=}3$	44.6	71.8	58.0	27.0	80.2	40.0	31.1	88.5	25.7	67.0	48.5	52.95
FIXED-K, $k{=}2$	28.7	51.0	50.6	21.4	24.6	0.0	0.0	22.9	1.1	22.1	27.3	22.70
FIXED-K, $k{=}1$	22.2	26.4	48.9	14.0	2.0	0.0	0.2	1.0	0.0	7.4	28.8	13.72
Qwen3-Next-80B-A3B-Instruct, natively $K{=}10$												
unpruned, $k{=}10$	63.1	86.9	76.2	34.2	88.4	79.7	67.1	95.2	53.1	82.1	74.2	72.75
FIXED-K, $k{=}9$	63.4	87.0	75.7	34.2	89.4	79.4	68.5	95.6	53.1	82.2	76.3	73.16
FIXED-K, $k{=}8$	63.1	86.5	74.5	34.6	90.0	81.8	68.5	95.8	54.9	82.4	68.7	72.80
FIXED-K, $k{=}7$	63.1	86.0	74.6	35.8	90.4	81.8	68.8	95.2	54.9	82.0	74.2	73.35
FIXED-K, $k{=}6$	63.1	85.6	73.9	33.6	90.8	82.2	66.6	95.7	54.9	82.1	76.3	73.16
FIXED-K, $k{=}5$	59.6	84.9	71.0	32.6	90.8	81.1	66.0	94.9	54.9	81.8	69.2	71.53
FIXED-K, $k{=}4$	55.9	82.2	67.2	30.2	88.0	74.8	58.8	93.1	49.1	80.7	72.7	68.43
FIXED-K, $k{=}3$	50.0	77.4	62.1	29.8	87.6	57.8	40.8	86.1	34.9	78.0	66.7	61.02
FIXED-K, $k{=}2$	36.8	61.1	56.1	22.2	69.0	12.0	7.7	75.0	7.4	62.3	44.4	41.27
FIXED-K, $k{=}1$	22.4	35.0	49.6	16.4	1.6	0.0	0.0	0.9	0.0	8.5	28.3	14.79
Ling-lite-1.5, natively $K{=}6$												
unpruned, $k{=}6$	60.2	82.3	70.3	31.6	89.2	40.9	28.1	91.4	34.3	69.4	56.6	59.48
FIXED-K, $k{=}5$	59.9	81.9	69.4	31.0	91.0	43.0	27.7	90.3	33.7	69.1	58.1	59.55
FIXED-K, $k{=}4$	58.3	81.2	66.1	30.8	89.0	38.7	26.0	90.5	33.7	67.4	55.6	57.94
FIXED-K, $k{=}3$	52.3	79.4	63.9	29.4	86.0	31.5	24.2	87.5	28.6	63.3	46.5	53.87
FIXED-K, $k{=}2$	38.9	69.6	57.4	23.0	70.4	5.0	8.4	72.4	8.0	42.3	38.4	39.44
FIXED-K, $k{=}1$	21.0	29.0	48.5	15.2	1.2	0.0	0.2	1.0	0.0	7.5	27.8	13.76
GPT-OSS-20B, natively $K{=}4$												
unpruned, $k{=}4$	45.3	77.5	66.9	27.2	90.6	73.7	71.7	86.2	58.3	74.3	64.6	66.94
FIXED-K, $k{=}3$	44.8	76.7	65.4	27.4	91.0	77.4	74.5	87.0	61.7	73.6	65.7	67.74
FIXED-K, $k{=}2$	44.3	75.9	64.1	26.0	86.6	73.9	69.4	88.7	48.0	70.5	65.2	64.79
FIXED-K, $k{=}1$	23.8	45.0	50.2	18.0	13.2	0.4	0.6	7.6	1.1	11.9	25.3	17.92

D FULL RESULTS: EVERY RULE AT BOTH BUDGETS

Tables 7 to 10 expand Table 3. For each core model they give every rule we evaluated at each of the two budget tiers, per dataset, with FIXED-K at the head of each tier so that the comparison the main text summarises can be read directly. Table 3 reports the best rule per tier; these tables report all of them, which is what supports two claims made in passing in §5.2. The first is that the ordering of rules changes between the two tiers on the same model. The second is that the spread between rules at a matched budget is wider than the gap between the best rule and FIXED-K, so a comparison that reports one rule against an unpruned baseline, without FIXED-K at the same $\bar{k}$, is not measuring what it appears to measure.

Budgets are the measured $\bar{k}$ of the run, generative and QA solved separately (§A.4). A rule missing from a tier either has no configuration that lands within 10% of it or did not complete; those cases are not silently replaced by a cheaper configuration.

E FULL RESULTS: THE MATCHED PAIRS

Table 11 gives the per-dataset FIXED-K sweeps behind the two panels of Figure 3, and Tables 12 and 13 the two sides of Figure 4. The reference checkpoint appears in both so that each pair can be read on its own.

Table 6: Uniform truncation, every budget and every dataset, on the five further architectures. Each row is one value of k applied to all tokens and all layers; the top row of each block is the unpruned model. Scores are each dataset's primary metric as a percentage and the last column is the unweighted mean over the datasets that were run: a dash marks a cell that was not, and it is left out of that row's mean. Green rows are above half the native K, where uniform truncation needs no rule to help it; orange rows are at or below it, past the crossover of §5. Columns, tints and the reference row follow Table 5.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
MiniMax-M2.7, natively $K{=}8$												
unpruned, $k{=}8$	70.1	86.8	76.2	38.8	90.2	81.5	81.7	91.6	47.3	82.0	85.4	75.60
FIXED-K, $k{=}7$	69.7	86.3	74.6	37.8	89.6	80.8	82.1	92.0	44.0	81.9	87.9	75.15
FIXED-K, $k{=}6$	69.2	85.6	74.4	37.4	88.8	78.3	80.6	91.5	42.3	81.5	84.3	73.99
FIXED-K, $k{=}5$	66.3	84.3	70.0	37.0	89.0	73.2	75.8	91.7	44.6	80.9	84.3	72.46
FIXED-K, $k{=}4$	63.1	82.9	68.0	35.8	85.4	54.2	42.9	90.1	33.1	78.7	78.8	64.82
FIXED-K, $k{=}3$	52.3	75.9	60.5	32.6	65.8	4.2	10.8	83.6	9.1	59.4	51.0	45.93
FIXED-K, $k{=}2$	34.9	53.8	53.2	22.2	3.0	0.0	0.0	2.3	0.0	9.4	24.2	18.45
FIXED-K, $k{=}1$	20.6	26.1	49.6	15.6	1.4	0.0	0.0	2.0	0.0	5.2	23.2	13.06
DeepSeek-V2-Lite-Chat, natively $K{=}6$												
unpruned, $k{=}6$	50.9	80.1	71.4	36.2	25.0	—	—	63.1	10.3	27.4	28.3	43.63
FIXED-K, $k{=}5$	50.3	79.4	70.8	35.6	24.6	—	—	61.7	10.9	28.2	30.3	43.53
FIXED-K, $k{=}4$	50.3	79.1	70.0	35.4	24.0	—	—	61.7	10.3	26.9	30.8	43.17
FIXED-K, $k{=}3$	47.6	77.4	69.9	33.8	26.2	—	—	56.1	6.9	24.3	26.3	40.94
FIXED-K, $k{=}2$	43.1	73.4	67.6	28.8	16.6	—	—	47.8	5.7	18.8	25.8	36.40
FIXED-K, $k{=}1$	33.6	63.7	61.8	25.2	3.2	—	—	14.9	1.1	11.1	25.3	26.66
Hy3, natively $K{=}8$												
unpruned, $k{=}8$	73.4	88.3	80.1	50.6	81.4	93.8	92.1	95.8	67.4	86.5	85.3	81.34
FIXED-K, $k{=}7$	72.9	88.1	78.5	50.4	81.0	93.3	89.8	95.8	65.7	86.6	84.3	80.58
FIXED-K, $k{=}6$	71.2	87.6	77.1	49.0	81.6	92.5	87.9	96.0	64.0	86.6	82.8	79.68
FIXED-K, $k{=}5$	69.9	86.0	76.2	47.6	80.8	90.6	82.1	95.5	57.7	86.5	81.8	77.70
FIXED-K, $k{=}4$	66.1	84.8	73.6	45.2	80.2	82.5	67.9	95.3	48.0	86.1	79.8	73.60
FIXED-K, $k{=}3$	61.4	79.8	67.6	44.0	74.0	32.3	25.6	94.7	24.0	83.9	60.1	58.85
FIXED-K, $k{=}2$	46.5	66.1	58.1	34.2	1.4	0.0	0.0	—	0.0	—	24.8	25.67
FIXED-K, $k{=}1$	25.4	31.1	51.0	27.8	1.6	0.0	0.0	0.8	0.0	5.8	25.2	15.33
DeepSeek-V4-Flash-0731, natively $K{=}6$												
unpruned, $k{=}6$	67.3	87.8	78.8	47.8	88.8	92.1	91.7	75.0	29.7	86.3	87.4	75.69
FIXED-K, $k{=}5$	66.2	86.8	78.1	48.8	88.0	92.9	93.5	76.8	32.0	86.7	86.9	76.07
FIXED-K, $k{=}4$	62.2	85.2	75.0	46.0	85.8	92.1	92.9	75.7	28.6	86.5	89.9	74.53
FIXED-K, $k{=}3$	58.0	81.7	70.8	44.6	85.4	91.0	89.2	81.7	29.1	85.3	86.4	73.02
FIXED-K, $k{=}2$	48.8	70.5	63.1	37.8	77.8	74.4	68.5	88.7	44.0	75.3	72.7	65.61
FIXED-K, $k{=}1$	27.6	39.3	52.3	27.2	1.2	0.0	0.0	1.7	0.0	6.1	12.6	15.27
Gemma 4 26B-A4B, natively $K{=}8$												
unpruned, $k{=}8$	45.6	67.2	56.7	31.6	85.2	91.5	86.9	92.2	78.9	84.6	80.3	72.77
FIXED-K, $k{=}7$	44.3	65.1	56.2	31.0	85.0	92.1	87.5	92.3	76.6	84.9	79.3	72.20
FIXED-K, $k{=}6$	42.6	63.6	53.9	31.6	84.6	89.8	85.4	92.2	72.0	84.5	79.8	70.92
FIXED-K, $k{=}5$	39.1	59.9	53.0	28.8	84.8	88.8	84.2	92.5	73.7	84.0	77.3	69.64
FIXED-K, $k{=}4$	35.8	51.3	52.6	29.4	85.6	84.6	71.9	91.5	73.1	82.5	72.7	66.45
FIXED-K, $k{=}3$	32.8	45.0	52.2	31.0	82.6	51.5	41.2	79.8	62.9	77.3	69.7	56.90
FIXED-K, $k{=}2$	29.7	37.4	52.0	29.4	40.6	2.1	0.2	47.5	20.0	55.0	41.4	32.30
FIXED-K, $k{=}1$	25.0	29.3	50.7	31.2	3.6	0.0	0.0	2.5	0.0	10.7	30.3	16.66

The two suites of Figure 4 are matched in size rather than in content. The multimodal side is MMMU (Yue et al., 2023), MMBench (Liu et al., 2023), MMStar (Chen et al., 2024), ChartQA (Masry et al., 2022), InfographicVQA (Mathew et al., 2021), TextVQA (Singh et al., 2019), GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018) and MME-RealWorld (Zhang et al., 2024b). The text side keeps the four likelihood-scored datasets of §3.3 and adds BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2019), Social IQa (Sap et al., 2019), MMLU (Hendrycks et al., 2020) and HellaSwag (Zellers et al., 2019), so that both models are read on nine short-answer datasets apiece.

The figures average the four hardest generative datasets, which is a choice made for legibility; these tables allow the same pairs to be read on any column. Two things visible here that the figures compress are worth pointing at. The advantage that reasoning-mode post-training and parameter scale buy at three experts is concentrated in the columns that need a long derivation, and is absent or slightly negative on the four likelihood-scored ones, which is the split §6 attributes it to. And the multimodal collapse below four experts is not uniform across the multimodal suite either. ChartQA falls furthest at every budget, and at two experts the three datasets that require transcribing text out

Table 7: Every rule at both budget tiers on **Qwen3-30B-A3B** (K=8), per dataset. Uniform truncation is the first row of each tier so that the two can be read together. The budget beside a rule is its measured $\bar{k}$, generative / QA. A dagger marks a configuration that spends more than 10% over its tier: it is shown for completeness but is not an iso-budget comparison and was not eligible for Table 3. The best eligible mean is bold where it beats FIXED-K, which is the row Table 3 carries.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
unpruned, k=8	60.8	85.0	73.2	32.0	89.6	74.1	61.4	94.4	41.1	74.2	56.6	67.49
Conservative budget, $\bar{k} \approx 5$ (ρ=0.62)												
FIXED-K, k=5	54.9	81.4	69.5	31.0	91.0	72.1	55.4	93.9	41.7	73.8	55.0	65.43
NAEE, 5.13/5.04	51.5	80.3	70.4	32.6	81.0	69.1	53.6	92.6	43.4	72.5	56.1	63.92
DIEP, 4.84/5.07	52.6	79.1	67.2	31.2	81.8	35.3	26.5	92.7	14.9	70.2	46.0	54.32
DIEP-d, 4.93/5.08	51.8	79.2	69.1	30.8	89.4	60.9	49.7	92.5	32.0	71.2	51.0	61.60
DYNROUTE, 4.91/5.09	56.5	81.4	71.6	31.0	89.6	73.2	55.2	94.2	41.7	73.2	55.1	**65.70**
BAN, 5.00/5.03	55.3	81.8	69.7	30.6	90.2	72.0	56.5	94.2	39.4	73.9	55.6	65.38
Aggressive budget, $\bar{k} \approx 3$ (ρ=0.38)												
FIXED-K, k=3	44.6	71.8	58.0	27.0	80.2	40.0	31.1	88.5	25.7	67.0	48.5	52.95
NAEE, 2.99/3.06	41.3	68.4	59.7	27.6	75.6	23.2	16.9	81.1	13.7	57.4	39.4	45.85
DIEP, 3.36/3.07†	38.9	67.6	54.0	23.8	58.4	3.3	6.0	86.4	4.6	59.5	39.9	40.22
DIEP-d, 3.01/3.04	39.3	68.8	56.8	25.0	62.8	5.1	6.9	82.3	4.0	56.9	38.4	40.57
DYNROUTE, 2.97/3.08	45.4	72.9	61.2	27.6	81.8	36.0	27.5	82.6	22.3	66.1	49.0	52.04
BAN, 3.14/3.08	46.1	72.8	61.5	25.8	88.0	42.9	34.0	90.4	28.6	68.8	50.5	**55.40**

Table 8: Every rule at both budget tiers on **Qwen3-Next-80B** (K=10), per dataset. Uniform truncation is the first row of each tier so that the two can be read together. The budget beside a rule is its measured $\bar{k}$, generative / QA. A dagger marks a configuration that spends more than 10% over its tier: it is shown for completeness but is not an iso-budget comparison and was not eligible for Table 3. The best eligible mean is bold where it beats FIXED-K, which is the row Table 3 carries.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
unpruned, k=10	63.1	86.9	76.2	34.2	88.4	79.7	67.1	95.2	53.1	82.1	74.2	72.75
Conservative budget, $\bar{k} \approx 6$ (ρ=0.60)												
FIXED-K, k=6	63.1	85.6	73.9	33.6	90.8	82.2	66.6	95.7	54.9	82.1	76.3	73.16
NAEE, 6.16/6.07	59.8	83.6	72.0	32.6	89.6	80.1	63.0	95.1	49.1	81.2	69.2	70.48
DIEP, 6.24/6.13	61.1	85.2	73.6	32.6	89.8	77.4	62.1	95.0	48.0	81.5	74.7	71.00
DIEP-d, 6.40/6.15	60.3	84.9	73.4	31.8	90.4	79.1	64.0	95.0	53.1	81.3	72.7	71.45
DYNROUTE, 5.97/6.20	62.2	85.7	73.0	35.6	89.6	83.4	67.9	95.1	50.9	81.9	77.3	72.96
BAN, 6.18/6.08	62.4	85.7	74.1	32.4	89.2	82.7	69.7	95.3	54.9	82.2	74.2	72.98
Aggressive budget, $\bar{k} \approx 4$ (ρ=0.40)												
FIXED-K, k=4	55.9	82.2	67.2	30.2	88.0	74.8	58.8	93.1	49.1	80.7	72.7	68.43
NAEE, 4.17/4.05	53.0	79.4	66.1	29.8	90.4	70.9	53.4	93.2	42.3	78.7	70.7	66.17
DIEP-d, 4.20/4.08	52.4	80.2	66.8	31.0	92.4	71.7	53.0	94.1	42.9	80.1	67.2	66.53
DYNROUTE, 4.30/4.08	57.8	82.3	70.8	33.0	89.2	78.5	62.2	95.0	54.3	81.1	69.7	70.35
BAN, 4.27/4.18	59.8	83.5	67.8	32.8	88.6	79.4	64.6	94.4	57.1	81.3	73.2	**71.14**

of an image are at or near zero, while the multiple-choice sets still return between a third and a half of their unpruned scores. The pattern is the one §4.2 describes, appearing inside a single suite: what breaks first is the requirement to emit an exact string.

F THROUGHPUT MEASUREMENTS

We time both backends through the generation call a user would make, not through a kernel benchmark. Each measurement fixes a batch size, a prompt length and a number of generated tokens, and every configuration measured at that shape sees the same random prompts. Generation runs to the requested length rather than stopping early, and prefix caching is disabled in vLLM so that one measurement cannot reuse work from another.

The two phases are timed separately by differencing rather than by instrumenting the backends. A run with a single generated token gives prefill alone; the decode rate for a longer run at the same batch size and prompt length is the remaining generated tokens divided by its wall time minus the matching prefill time. Every run is executed once to warm up and then timed repeatedly at a fixed seed, and the reported rate uses the median across trials.

Table 9: Every rule at both budget tiers on **Ling-lite-1.5** (K=6), per dataset. Uniform truncation is the first row of each tier so that the two can be read together. The budget beside a rule is its measured $\bar{k}$, generative / QA. A dagger marks a configuration that spends more than 10% over its tier: it is shown for completeness but is not an iso-budget comparison and was not eligible for Table 3. The best eligible mean is bold where it beats FIXED-K, which is the row Table 3 carries.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
unpruned, k=6	60.2	82.3	70.3	31.6	89.2	40.9	28.1	91.4	34.3	69.4	56.6	59.48
Conservative budget, $\bar{k} \approx 4$ (ρ=0.67)												
FIXED-K, k=4	58.3	81.2	66.1	30.8	89.0	38.7	26.0	90.5	33.7	67.4	55.6	57.94
NAEE, 4.06/4.24	56.3	81.5	65.7	29.4	87.4	33.2	23.3	87.9	30.3	61.9	53.5	55.49
DIEP, 4.09/4.08	56.8	81.1	64.4	29.8	87.4	35.1	24.3	88.7	28.0	63.5	52.0	55.55
DIEP-d, 3.98/4.32	57.4	81.6	67.8	30.4	86.2	34.1	25.8	87.9	32.0	62.8	58.1	56.74
DYNROUTE, 4.17/4.22	58.4	82.0	68.4	30.4	90.2	40.1	26.8	90.5	32.0	67.3	58.6	**58.61**
BAN, 3.97/4.07	58.0	81.7	68.4	30.6	89.2	39.8	25.9	89.8	34.9	67.8	55.1	58.29
Aggressive budget, $\bar{k} \approx 2$ (ρ=0.33)												
FIXED-K, k=2	38.9	69.6	57.4	23.0	70.4	5.0	8.4	72.4	8.0	42.3	38.4	39.44
NAEE, 2.43/2.28†	46.9	73.7	57.2	26.4	74.2	12.0	11.9	78.8	18.9	48.5	46.5	45.00
DIEP, 2.69/2.25†	43.9	73.1	57.7	25.8	77.0	15.1	15.9	84.1	17.1	51.1	40.9	45.61
DIEP-d, 2.08/2.05	43.7	69.8	57.8	24.0	68.6	6.4	9.9	75.1	10.9	42.4	32.3	40.08
DYNROUTE, 2.04/2.10	44.5	69.5	56.6	25.4	65.2	4.2	7.8	66.8	8.0	38.3	32.8	38.10
BAN, 2.05	42.2	71.2	57.1	24.6	70.4	10.2	11.5	75.1	14.9	47.5	41.9	**42.42**

Table 10: Every rule at both budget tiers on **GPT-OSS-20B** (K=4), per dataset. Uniform truncation is the first row of each tier so that the two can be read together. The budget beside a rule is its measured $\bar{k}$, generative / QA. A dagger marks a configuration that spends more than 10% over its tier: it is shown for completeness but is not an iso-budget comparison and was not eligible for Table 3. The best eligible mean is bold where it beats FIXED-K, which is the row Table 3 carries.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
unpruned, k=4	45.3	77.5	66.9	27.2	90.6	73.7	71.7	86.2	58.3	74.3	64.6	66.94
Conservative budget, $\bar{k} \approx 3$ (ρ=0.75)												
FIXED-K, k=3	44.8	76.7	65.4	27.4	91.0	77.4	74.5	87.0	61.7	73.6	65.7	67.74
NAEE, 2.97	40.9	73.9	66.0	26.4	88.8	73.5	69.0	87.2	58.3	71.1	63.1	65.28
DIEP, 2.86	39.2	73.0	63.9	22.4	85.2	50.4	44.2	86.2	32.0	67.7	57.1	56.47
DIEP-d, 2.99	42.8	75.9	67.6	27.6	91.6	72.4	65.9	85.3	55.4	71.7	66.2	65.67
DYNROUTE, 2.97	43.4	77.8	65.1	28.2	90.0	77.0	72.4	86.6	62.3	73.5	63.1	67.21
BAN, 3.05	44.8	76.8	66.2	28.0	92.4	73.1	70.4	86.9	60.0	73.4	60.6	66.59
Aggressive budget, $\bar{k} \approx 2$ (ρ=0.50)												
FIXED-K, k=2	44.3	75.9	64.1	26.0	86.6	73.9	69.4	88.7	48.0	70.5	65.2	64.79
NAEE, 2.37†	36.8	68.3	61.4	23.0	87.2	76.0	71.1	87.8	53.7	71.8	65.7	63.90
DIEP, 2.52†	34.0	63.9	58.5	19.4	85.4	74.1	69.6	88.6	49.7	71.9	65.2	61.85
DIEP-d, 2.31†	42.7	77.6	63.8	25.2	88.2	74.3	70.4	88.3	49.1	70.9	61.6	64.74
DYNROUTE, 2.07	42.3	73.6	63.1	26.1	89.9	73.7	71.2	85.6	62.5	70.8	63.6	**65.69**
BAN, 1.99	44.8	76.2	65.0	26.2	86.2	71.5	66.0	87.2	40.6	68.2	63.1	63.18

FIXED-K is applied here without the router patch of Appendix A. We write a copy of the model configuration with a smaller number of experts per token, so the serving stack selects fewer experts natively and follows the same code path it would for a model published at that width. The speedups therefore carry none of the overhead that a per-token allocation rule would add, which is the distinction §4.3 draws between what the budget buys and what a rule would cost to execute.

G MODALITY AND ROUTING CONCENTRATION

§6 compares reasoning mode and parameter scale, both of which separate their pair only once pruning turns aggressive. Modality behaves the same way, and the router explains why.

Figure 4 compares Qwen3-VL-30B-A3B-Instruct with Qwen3-30B-A3B-Instruct on nine datasets each: multimodal tasks for the former and likelihood-scored QA for the latter. Their normalised curves remain within one percentage point from eight to four experts. At three experts, the multimodal model falls 11 points behind; at two, the gap widens to 42 points. The similar response to conservative pruning gives way to sharply different losses as the budget tightens.

Table 11: The two matched pairs of Figure 3 in full, per dataset. The reference checkpoint is repeated from Table 5 so that each pair can be read without turning back, and the budget tints are the same as in that table. Two budgets are not reported, $k{=}1$ for Qwen3-30B-A3B-Thinking and $k{=}2$ for Qwen3-235B-A22B-Instruct: at these budgets generation has essentially collapsed and fails to terminate normally, which makes the generative runs prohibitively long. Figure 3 averages the four hardest generative columns, AIME-24, AIME-25, LCB and GPQA-D.

	Knowledge QA				Math & code reasoning					Knowledge & general		
	ARC-c	ARC-e	WG	OBQA	MATH	AIME24	AIME25	GSM8K	LCB	MMLU-P	GPQA-D	Avg
Qwen3-30B-A3B-Instruct (reference), natively $K{=}8$												
unpruned, $k{=}8$	60.8	85.0	73.2	32.0	89.6	74.1	61.4	94.4	41.1	74.2	56.6	67.49
FIXED-K, $k{=}7$	60.3	84.0	73.2	33.2	89.2	74.6	61.3	94.7	41.3	74.5	56.6	67.54
FIXED-K, $k{=}6$	59.2	82.8	71.3	31.4	90.4	72.8	59.3	95.0	44.6	74.4	53.5	66.79
FIXED-K, $k{=}5$	54.9	81.4	69.5	31.0	91.0	72.1	55.4	93.9	41.7	73.8	55.0	65.43
FIXED-K, $k{=}4$	51.3	77.2	65.1	31.2	87.8	62.5	47.9	93.3	40.6	72.3	54.0	62.11
FIXED-K, $k{=}3$	44.6	71.8	58.0	27.0	80.2	40.0	31.1	88.5	25.7	67.0	48.5	52.95
FIXED-K, $k{=}2$	28.7	51.0	50.6	21.4	24.6	0.0	0.0	22.9	1.1	22.1	27.3	22.70
FIXED-K, $k{=}1$	22.2	26.4	48.9	14.0	2.0	0.0	0.2	1.0	0.0	7.4	28.8	13.72
Qwen3-30B-A3B-Thinking, natively $K{=}8$												
unpruned, $k{=}8$	55.6	83.1	72.5	29.8	92.8	88.0	81.5	95.5	64.6	80.8	69.2	73.95
FIXED-K, $k{=}7$	55.2	82.2	72.1	30.0	93.2	88.1	80.4	95.7	62.9	80.9	70.2	73.72
FIXED-K, $k{=}6$	54.4	81.1	69.2	30.4	91.8	86.9	80.6	95.5	65.7	80.7	74.7	73.73
FIXED-K, $k{=}5$	51.2	78.3	67.9	29.4	93.4	84.7	78.5	95.8	59.4	80.4	70.2	71.75
FIXED-K, $k{=}4$	45.9	75.0	64.2	28.4	94.0	79.2	73.9	95.6	54.9	79.5	66.7	68.85
FIXED-K, $k{=}3$	40.1	67.8	56.3	28.0	90.0	61.8	50.1	93.3	30.3	75.1	62.1	59.54
FIXED-K, $k{=}2$	28.2	48.6	53.1	20.8	28.0	0.1	0.0	33.1	0.0	25.9	23.7	23.77
Qwen3-235B-A22B-Instruct, natively $K{=}8$												
unpruned, $k{=}8$	59.8	83.4	75.4	35.2	88.4	83.3	71.9	94.4	49.7	78.8	63.1	71.22
FIXED-K, $k{=}7$	59.6	82.6	74.7	33.2	89.2	80.0	69.0	94.9	49.7	78.4	63.6	70.45
FIXED-K, $k{=}6$	57.3	80.8	73.2	33.2	91.4	80.8	68.5	93.7	49.7	77.5	62.1	69.84
FIXED-K, $k{=}5$	56.7	80.3	70.6	32.4	93.2	78.3	66.9	93.4	52.0	77.5	60.1	69.22
FIXED-K, $k{=}4$	52.2	76.5	66.0	33.0	91.4	70.8	61.9	90.1	52.0	76.2	57.6	66.15
FIXED-K, $k{=}3$	42.1	68.3	58.0	29.6	86.4	56.9	48.8	89.2	38.3	72.6	61.6	59.25
FIXED-K, $k{=}1$	23.7	27.8	50.7	14.2	1.8	0.2	0.0	1.5	0.0	6.2	28.8	14.08

Table 12: Uniform truncation on the model behind the multimodal side of Figure 4, per dataset. The two sides of that pair are nine datasets each and both are short answers, but they are not the same nine, so they are comparable only through the mean and only relative to each model's own unpruned row. Tints are the budget regions of Table 5. Seven of the nine sets are the `lite` subsets.

Qwen3-VL-30B-A3B-Instruct, nine multimodal datasets, natively $K{=}8$										
	MMMU	MMB	MMStar	ChartQA	InfoVQA	TextVQA	GQA	VizWiz	MME-RW	Avg
unpruned, $k{=}8$	58.4	88.6	71.7	78.4	80.6	84.4	72.2	59.1	52.7	71.79
FIXED-K, $k{=}7$	58.2	90.2	71.1	76.4	79.1	84.8	71.4	58.7	54.2	71.57
FIXED-K, $k{=}6$	57.6	90.2	71.2	71.8	78.9	82.9	70.6	60.1	54.0	70.81
FIXED-K, $k{=}5$	56.0	88.6	69.6	68.4	78.5	82.8	70.6	58.8	52.9	69.58
FIXED-K, $k{=}4$	49.9	90.2	65.0	55.4	73.4	79.3	66.8	56.3	50.9	65.24
FIXED-K, $k{=}3$	47.3	83.3	55.3	28.6	56.0	58.7	54.6	44.3	39.9	52.00
FIXED-K, $k{=}2$	29.2	37.9	25.3	0.4	2.9	2.5	8.2	1.2	20.2	14.20
FIXED-K, $k{=}1$	26.2	15.9	23.3	0.0	0.0	0.0	0.0	0.0	19.7	9.46

Routing statistics reveal a corresponding difference in weight concentration. We record each router's full 128-way softmax at every one of the 48 MoE layers, before top-K selection, while the two models run the same suites as above, and aggregate the distributions over evaluated tokens. Both models are measured on prompt tokens, the comparable stage across the pair. Within the selected eight experts, the leading three hold 49.8% of the multimodal model's routed weight, compared with 59.8% for the text model. Truncating to three therefore discards 50.2% and 40.2% of the selected weight, respectively, before renormalisation. The same retained expert count removes a larger share of routed weight in the multimodal model.

This difference extends to the full routing distribution: the selected eight experts hold 15.5% of the 128-way softmax mass in the multimodal model and 44.3% in the text model. Per-dataset values do not overlap on either concentration measure, and the direction is consistent across all 48 MoE layers. The flatter routing distribution helps explain why aggressive truncation is more disruptive in the multimodal evaluation.

Table 13: Uniform truncation on the model behind the text side of Figure 4, per dataset. The two sides of that pair are nine datasets each and both are short answers, but they are not the same nine, so they are comparable only through the mean and only relative to each model's own unpruned row. Tints are the budget regions of Table 5. The five datasets after OpenBookQA were measured for this comparison alone and are not part of the suite of §3.3.

Qwen3-30B-A3B-Instruct, nine likelihood-scored QA datasets, natively $K{=}8$										
	ARC-c	ARC-e	WG	OBQA	BoolQ	PIQA	SIQA	MMLU	HSwag	Avg
unpruned, $k{=}8$	60.8	85.0	73.2	32.0	88.7	80.1	54.1	80.1	61.2	68.36
FIXED-K, $k{=}7$	60.3	84.0	73.2	33.2	88.3	79.7	53.8	80.0	61.1	68.18
FIXED-K, $k{=}6$	59.2	82.8	71.3	31.4	86.7	79.6	53.5	79.2	60.9	67.18
FIXED-K, $k{=}5$	54.9	81.4	69.5	31.0	84.7	78.7	51.1	78.3	60.2	65.53
FIXED-K, $k{=}4$	51.3	77.2	65.1	31.2	82.1	76.6	47.6	75.3	58.4	62.76
FIXED-K, $k{=}3$	44.6	71.8	58.0	27.0	75.0	72.9	44.3	67.5	52.7	57.09
FIXED-K, $k{=}2$	28.7	51.0	50.6	21.4	53.8	62.0	36.6	34.4	39.9	42.04
FIXED-K, $k{=}1$	22.2	26.4	48.9	14.0	45.6	51.5	33.5	23.8	25.9	32.42

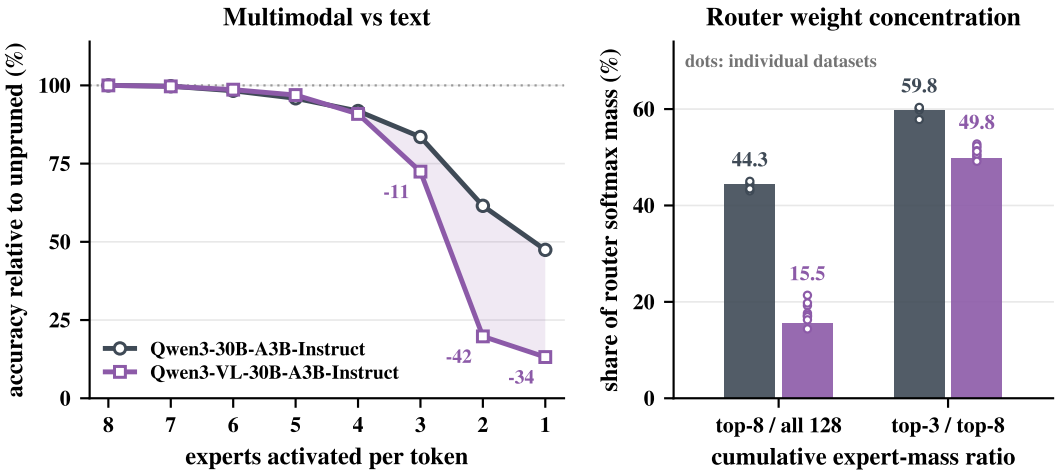

Figure 4: Modality, measured on the scores and on the router. Left: uniform truncation on nine datasets a side, the multimodal suite against nine likelihood-scored QA datasets on the text model, each curve normalised to its own unpruned score; annotations give the gap in points. Right: the router's full 128-way softmax, recorded before top-k selection over all 48 MoE layers and weighted by token count. Dots are the individual datasets.

Recording concentration on prompt tokens raises the question of whether the multimodal router's flatness belongs to the model or to the stage. The VL model's own generated tokens answer it. Over its decode stage the natively selected eight experts hold 20.5% of the full softmax and the leading three hold 51.7% of those eight, against 15.5% and 49.8% on its prompts. Its routing is therefore somewhat more peaked when generating than when reading, but still far flatter than the text model's 44.3% and 59.8%. The difference between the two models is a property of the models rather than of the stage at which we record it.

The size of that difference is also stable across depth. A 1000-sample bootstrap over layers puts the gap in top-eight mass at -26.0 points with a 95% interval of $[-31.1, -21.2]$, and the router's entropy at $+0.70$ nats higher in the multimodal model with an interval of $[+0.54, +0.85]$.