CoCurve: Cross-Module Co-Pruning Curvature for Structured LLM Pruning

Anonymous authors
Paper under double-blind review

Abstract

Resource-constrained deployment requires sustaining large language model (LLM) capabilities as models scale under fixed memory and computation budgets. Structured pruning advances this deployment frontier with smaller dense checkpoints, yet deciding what to prune remains bottlenecked by interactions among jointly removed units. We introduce Cross-Module Co-Pruning Curvature (CoCurve), which formulates structured pruning as set-dependent predictive risk over a unified inventory of attention heads and feed-forward groups. A co-pruning graph built from single-unit forward ablations assigns individual risk to nodes and reinforcement or cancellation to interaction edges, conditioning each decision on the units already removed. We evaluate 6 LLMs (3B–70B) and 3 vision–language models (VLMs) across 3 perplexity corpora, 12 language tasks, and 7 multimodal benchmarks. Across the five-model 20–40% grid and the 70B 10–50% sweep, CoCurve ranks first in $53/60$ corpus comparisons ($15/15$ at 70B) and permits 2.2–6.6 more pruning points at matched quality in $9/10$ interpolated comparisons, while leading all 6 VLM Avg_7 blocks. After lightweight recovery, its retained structures remain strongest through 50% pruning, where the 8B checkpoint regains 10.8 Avg_{12} points; physical slicing delivers $1.58\times$ dense prefill throughput with 41% lower peak memory. Mechanism analysis across 10 LLMs and 7 VLMs further finds organized within- and cross-module edge structure; matched removals of low-saliency, high-coupling units degrade $19/20$ capability groups by up to 23.7 points. Code is included in the supplementary material.

1 Introduction

Across architectures and model scales, large language models (LLMs) and multimodal extensions support increasingly broad capabilities in language modeling and downstream tasks, yet their memory, latency, and computation demands complicate deployment on heterogeneous hardware (Grattafiori et al., 2024; Bai et al., 2025b). The practical target is therefore the *quality–deployment frontier*: how well these capabilities are retained within a resource budget. Moving this frontier is central to democratizing LLM deployment.

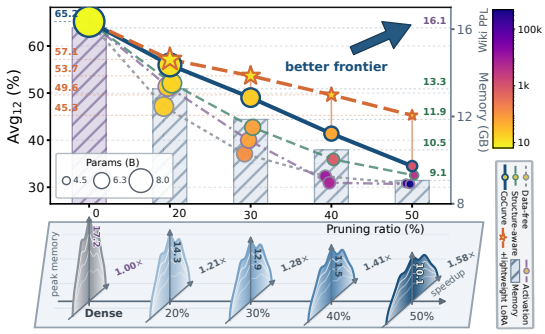

Figure 1: **CoCurve advances the quality–deployment frontier** on Llama-3.1-8B-Instruct.

Model compression advances this frontier through complementary routes. Quantization reduces memory and arithmetic but realizes speedups only through suitable low-precision kernels (Frantar et al., 2023), while distillation incurs extra training to produce a smaller model (Hinton et al., 2015; Muralidharan et al., 2024). Pruning instead removes pretrained computation directly. Unstructured pruning offers fine-grained removal but relies on sparse kernels or hardware to translate sparsity into speedups (Frantar & Alistarh, 2023; Sun et al., 2024), whereas structured pruning yields smaller dense checkpoints for standard hardware (Ma et al., 2023; An et al., 2024; Ashkboos et al., 2024). Existing structured pruners exploit weights, activations, gradients, reconstruction, or learned allocation (Ling et al., 2024; Guo et al., 2025; Li et al., 2025; Wang et al., 2026), yet mostly

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.

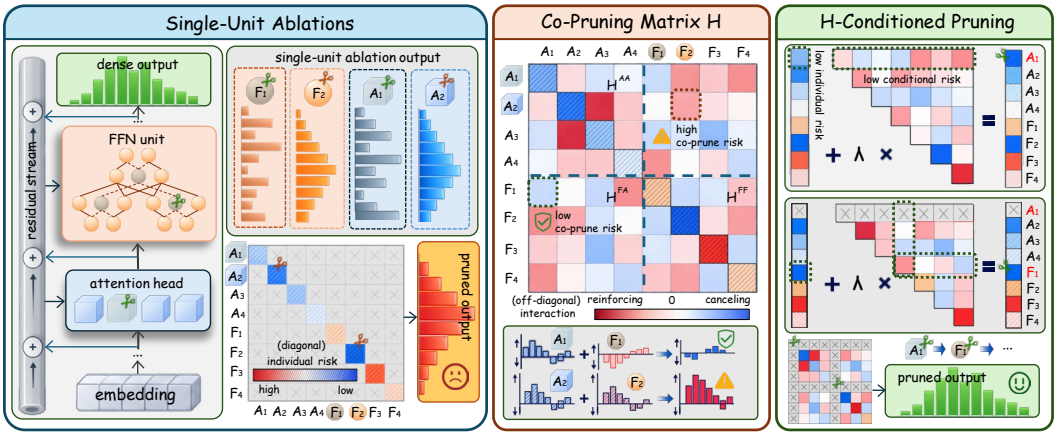

Figure 2: **CoCurve: from single-unit responses to graph-conditioned pruning.**

assign unit-wise scores, leaving how units interact when removed together unmodeled and limiting further gains on the quality–deployment frontier.

These interactions matter because structured pruning is inherently a joint-removal problem. Attention heads and feed-forward network (FFN) groups write to a shared residual stream (Vaswani et al., 2017; Elhage et al., 2021), through which their pruning effects can reinforce or partially cancel one another (McGrath et al., 2023; Gong et al., 2026b). In a mechanism analysis across five LLMs, we find that mean normalized attention–FFN edge magnitude is 0.74–$0.98\times$ the within-module mean (Figure 8a). We therefore ask: *How can interactions among structured-unit pruning effects inform which units to remove together that better preserve model quality?*

We introduce Cross-Module Co-Pruning Curvature (CoCurve), casting structured pruning as set-dependent predictive risk over a unified inventory of attention heads and FFN groups (Figure 2). A local expansion of teacher Kullback–Leibler (KL) risk separates diagonal *individual* risk from signed off-diagonal *interaction* edges, covering pairs within and across modules. A Fisher–Gram estimator builds this reusable graph from one forward ablation per unit, without gradients or weight updates. At each step, a conditional pruner updates each candidate's removal risk using its edges to the current removal set, and held-out predictive risk selects among the resulting masks. CoCurve thus revises risk as the removal set grows instead of applying a fixed unit ranking or module allocation.

We evaluate CoCurve on 6 LLMs (3B–70B) and 3 vision–language models (VLMs) over 3 perplexity corpora, 12 language tasks, and 7 multimodal benchmarks. Across the LLMs, CoCurve leads $53/60$ corpus comparisons on the five-model 20–40% grid and the 70B 10–50% sweep ($15/15$ at 70B), permits 2.2–6.6 points more pruning in $9/10$ matched-quality interpolations at 30–40%, and leads all 8 Avg_{12} blocks on the 8B/24B models. Across the VLMs, it leads every Avg_7 block, with 1.1–10.7-point margins at 30%. On the 8B LLM, its structure remains best under identical lightweight low-rank adaptation (LoRA) through 50% pruning, and under weight quantization. Physical slicing reaches $1.58\times$ dense prefill throughput with 41% lower peak memory, moving the frontier in Figure 1. Across 10 LLMs and 7 VLMs, mechanism analysis finds organized within- and cross-module edges that reverse an actual pruning decision; matched removals of low-saliency, high-coupling units degrade $19/20$ capability groups by up to 23.7 points. Thus modeling joint-removal interactions provides a direct route to advancing the quality–deployment frontier.

2 Method

Structured pruning is a set-selection problem: pruning risk depends not only on individual units but also on their removal set. Attention and FFN units meet in the shared residual stream, motivating one structured-pruning inventory $\mathcal{U} = \{u_1, \ldots, u_M\}$. With $s_u{=}1$ denoting pruning unit u, the goal is to meet a target parameter reduction while preserving quality and lowering deployment cost.

CoCurve formalizes this dependence as *set-dependent predictive risk*. It derives individual and pairwise terms (Sec. 2.1), estimates their graph from single-unit ablations (Sec. 2.2), and conditionally prunes with held-out-risk calibration (Sec. 2.3). Details and theory appear in Appendices A and B.

2.1 PREDICTIVE RISK DEFINES THE CO-PRUNING GRAPH

Building on Optimal Brain Damage (OBD), Optimal Brain Surgeon (OBS) (LeCun et al., 1989; Hassibi & Stork, 1992), and structured objectives for interacting removals (Yu et al., 2022; Nonnenmacher et al., 2022; Kwon et al., 2022), CoCurve evaluates all units through one predictive risk rather than separate module scores. For a calibration sequence x and token position t, let $p_0(\cdot \mid x_{\le t})$ and $p_s(\cdot \mid x_{\le t})$ denote the dense and gated next-token distributions, with gates relaxed continuously near the dense point $s = \mathbf{0}$. Forward teacher KL divergence uses the dense next-token distribution as unlabeled supervision, following the distribution-matching principle of knowledge distillation (Hinton et al., 2015). Its token-level predictive risk is

$$\mathcal{R}(s) = \mathbb{E}_{x,t}[D_{\mathrm{KL}}(p_0(\cdot \mid x_{\le t}) \,\|\, p_s(\cdot \mid x_{\le t}))]\,. \tag{1}$$

At the dense model, the distributions coincide, so $\mathcal{R}(\mathbf{0}) = 0$, $\nabla_s \mathcal{R}(\mathbf{0}) = 0$, and curvature is the first informative local term without assuming training-loss stationarity (Appendix B.1). For binary removal masks, $s_u^2 = s_u$, so the second-order Taylor approximation becomes

$$\mathcal{R}(s) \approx \tfrac{1}{2} s^\top \mathbf{H}^\star s = \tfrac{1}{2} \sum_u \mathbf{H}^\star_{uu} s_u + \sum_{u<v} \mathbf{H}^\star_{uv} s_u s_v, \qquad \mathbf{H}^\star = \nabla_s^2 \mathcal{R}(\mathbf{0}). \tag{2}$$

This decomposition naturally defines the *co-pruning graph*: diagonals encode individual pruning risk (nodes), while signed off-diagonals link removal pairs (edges). Positive edges capture reinforcement between Fisher-weighted logit responses and raise joint risk above isolated risks; negative edges capture partial cancellation and lower it (Proposition 10). Edges may link units within or across modules and layers; they measure alignment between pruning-induced predictive changes rather than only direct architectural adjacency. Thus the graph models set-dependent pruning risk beyond any static ranking.

2.2 SINGLE-UNIT ABLATIONS CONSTRUCT THE FULL GRAPH

The exact mask Hessian factors through unit-wise logit derivatives. Let z_0 and z_s denote the dense and gated next-token logits, write $\Delta z_s = z_0 - z_s$, and define $G_u = \partial \Delta z_s / \partial s_u|_{\mathbf{0}}$. At the dense model, the exact mask Hessian is $\mathbf{H}^\star_{uv} = \mathbb{E}_{x,t}[G_u^\top F_{x,t} G_v]$, where $F_{x,t} = \mathrm{diag}(p_0) - p_0 p_0^\top$ is the categorical Fisher matrix in logit space (Amari, 1998).

CoCurve replaces each derivative column with the finite next-token-logit response $\delta z_u = z_0 - z_{-u}$, where z_{-u} denotes the logits with only u pruned. Because every response occupies the same next-token-logit space, units from different layers and modules are directly comparable. Let $\bar{D}_u$ stack these responses after teacher-probability weighting and mean centering over P sampled positions. Their Gram matrix estimates every node and edge:

$$\mathbf{H}_{uv} := \tfrac{1}{P} \bar{D}_u^\top \bar{D}_v. \tag{3}$$

Thus $\mathbf{H}$ is positive semidefinite, and M single-unit forward ablations recover all $O(M^2)$ edges of the finite-ablation Gram surrogate without pairwise interventions, gradients, or weight updates (Proposition 3). On the 8B inventory, this is 768 rather than 294,528 model interventions.

2.3 ACCUMULATED EDGES CONDITION EACH PRUNING DECISION

Edges matter operationally when they change which unit is pruned next. To bridge local curvature and finite removal, CoCurve traces a positive-semidefinite path $\lambda \in [0, 1]$ from isolated risk to the full co-pruning graph; held-out predictive risk selects among the resulting masks:

$$\widehat{\mathcal{R}}_\lambda(s) = \tfrac{1}{2} \sum_u \mathbf{H}_{uu} s_u + \lambda \sum_{u<v} \mathbf{H}_{uv} s_u s_v = \tfrac{1}{2} s^\top \mathbf{H}_\lambda s, \quad \mathbf{H}_\lambda = (1-\lambda)\,\mathrm{diag}(\mathbf{H}) + \lambda \mathbf{H}. \tag{4}$$

At $\lambda = 0$ this recovers isolated saliency and at $\lambda = 1$ the full Gram surrogate; every intermediate matrix remains positive semidefinite. Here λ indexes candidate masks rather than measuring interaction strength: candidate u receives the accumulated signed correction $\lambda \sum_{v \in S} \mathbf{H}_{uv}$.

Let $c_u > 0$ be unit u's parameter cost, $C = \sum_u c_u$, $c_{\max} = \max_u c_u$, and $\mathcal{G} \subseteq \{0,1\}^M$ the masks satisfying fixed guards. For target fraction ρ, the constrained objective is

$$\min_{s \in \mathcal{G}} \widehat{\mathcal{R}}_\lambda(s) \quad \text{subject to} \quad \rho C \le \sum_u c_u s_u < \rho C + c_{\max}. \tag{5}$$

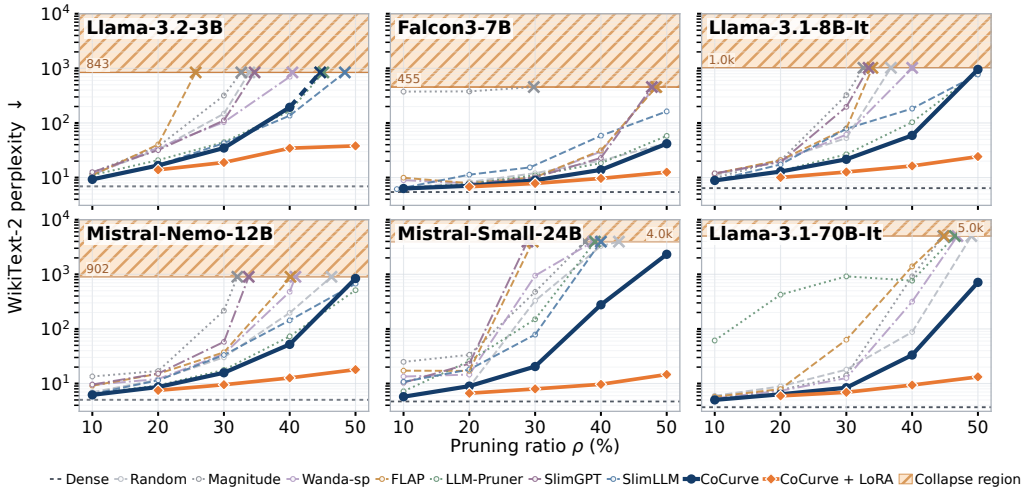

Figure 3: **WikiText-2 perplexity ↓ across pruning ratios.** Full sweep: Appendix D.1.1.

Attention and FFN units compete under this single objective, with no module-wise quota searched or prescribed. Exact minimization would search exponentially many heterogeneous-cost subsets, so CoCurve uses a cost-normalized conditional greedy rule. If S is the current removal set, the incremental surrogate risk of adding an eligible unit u is

$$\Delta\widehat{\mathcal{R}}_\lambda(u \mid S) = \tfrac{1}{2}\mathbf{H}_{uu} + \lambda \sum_{v \in S} \mathbf{H}_{uv}. \tag{6}$$

The first term is fixed individual saliency, while the second accumulates signed edges to the current set. We minimize $\Delta\widehat{\mathcal{R}}_\lambda(u \mid S)/c_u$ and stop at the target cost. Because the incident-edge sum changes with S, the ranking can change after every pruning step.

Closing the loop with held-out risk. Varying λ produces a finite family $\mathcal{F}$ of distinct removal sets (Proposition 7). On a disjoint split $\mathcal{C}'$, we evaluate each mask by its teacher-KL risk $\mathcal{R}(S;\mathcal{C}')$ and select $\widehat{S} = \arg\min_{S\in\mathcal{F}} \mathcal{R}(S;\mathcal{C}')$. The graph proposes candidate masks, while measured risk chooses the actual pruned model without downstream labels or weight updates.

3 Experiments

3.1 Evaluation setup

Models. We evaluate 6 *LLMs* from 3B to 70B: Llama-3.2-3B and Llama-3.1-Instruct at 8B/70B (Grattafiori et al., 2024); Falcon3-7B (Falcon-LLM Team, 2024); Mistral-Nemo-12B (Mistral AI Team, 2024); and Mistral-Small-24B (Mistral AI Team, 2025). We assess transfer to 3 *VLMs*: Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-8B (Bai et al., 2025a), and InternVL3-8B (Zhu et al., 2025). Mechanism analyses cover 10 LLMs (3B–14B) and 7 VLMs (2B–8B; Appendix E).

Evaluation. LLM evaluation covers 3 perplexity corpora and 12 zero-shot tasks grouped into commonsense reasoning, science QA, reading comprehension, and knowledge; VLM transfer covers 7 benchmarks across perception, reasoning, hallucination, and knowledge.

Baselines. The primary grid covers 3 signal classes: *data-free heuristics*, Random and Magnitude (Han et al., 2015); *activation statistics*, Wanda-sp (Sun et al., 2024) and FLAP (An et al., 2024); and *structure-aware optimization*, LLM-Pruner (Ma et al., 2023), SlimGPT (Ling et al., 2024), and SlimLLM (Guo et al., 2025). Appendix C.3 adds *allocation-aware* AMP (Mugnaini et al., 2025) and 2SSP (Sandri et al., 2025).

Protocol. All methods prune the same units under a shared parameter budget, layer guard, and harness, without weight updates or fine-tuning. CoCurve builds its graph on unlabeled data and selects $\hat{\lambda}$ on a disjoint split by teacher KL; downstream tasks never affect pruning. All efficiency metrics use one isolated A100-80GB. Appendix C details data, prompts, matching, and implementation.

Table 1: **Pruning-only quality across two model scales.** Family means and Avg_{12} are percentages; full task breakdowns appear in Appendix D.1.2.

		Llama-3.1-8B-Instruct						Mistral-Small-24B					
ρ	Method	Wiki↓	Cmn_4	Sci_3	Read_4	Know_1	Avg_{12}↑	Wiki↓	Cmn_4	Sci_3	Read_4	Know_1	Avg_{12}↑
–	**Dense**	6.4	70.9	60.1	67.6	47.8	65.2±0.34	4.7	73.5	63.4	65.9	52.3	66.7±0.33
20%	Random	21.2	53.8	37.0	49.1	31.8	46.2±0.34	**8.7**	64.6	53.0	58.0	43.2	57.7±0.34
	Magnitude	20.5	55.5	39.5	51.0	32.8	48.1±0.34	33.6	63.5	53.1	56.6	42.4	56.8±0.35
	Wanda-sp	20.0	59.3	45.9	52.6	35.6	51.7±0.35	14.6	62.7	51.2	58.6	41.0	56.6±0.35
	FLAP	20.9	58.4	46.1	51.4	35.3	51.1±0.35	17.0	61.6	50.1	57.1	40.6	55.5±0.35
	LLM-Pruner	13.2	62.2	44.8	58.5	36.8	54.5±0.34	25.8	65.5	55.2	59.8	**43.7**	59.2±0.34
	SlimGPT	19.2	58.3	44.9	48.3	35.1	49.7±0.35	23.1	64.3	52.0	57.5	41.4	57.0±0.35
	SlimLLM	17.4	59.8	**48.2**	51.1	37.2	52.1±0.35	18.1	67.0	54.6	58.8	43.5	59.2±0.34
	CoCurve	**12.9**	**63.7**	48.0	**58.6**	**38.5**	**56.0**±0.35	9.0	**67.8**	**56.3**	**60.9**	43.5	**60.6**±0.34
30%	Random	52.9	46.0	30.6	40.9	28.6	39.0±0.33	328.2	56.7	43.4	45.3	35.5	47.8±0.35
	Magnitude	324.5	41.2	27.4	37.0	27.3	35.2±0.33	477.5	50.1	39.0	46.8	31.7	44.7±0.34
	Wanda-sp	61.2	47.8	33.9	40.2	29.6	40.3±0.34	945.4	50.6	41.7	42.9	33.7	44.4±0.35
	FLAP	79.4	48.5	31.8	39.6	29.0	39.7±0.33	4.8e3	37.1	24.3	28.0	26.5	30.0±0.32
	LLM-Pruner	26.6	55.2	38.2	**52.1**	31.9	48.0±0.34	150.6	54.6	41.8	51.5	33.7	48.6±0.34
	SlimGPT	196.8	45.7	31.9	38.6	27.9	38.4±0.33	7.9e3	49.4	37.4	42.5	31.4	42.6±0.34
	SlimLLM	78.2	49.1	36.7	41.3	30.4	41.8±0.34	78.8	58.5	**49.8**	50.2	36.5	51.7±0.35
	CoCurve	**21.6**	**55.7**	**41.7**	51.8	**33.4**	**49.0**±0.35	**20.5**	**59.3**	48.2	**54.1**	**36.5**	**52.9**±0.35
40%	Random	4.2e3	39.4	26.2	34.4	26.7	33.4±0.32	3.3e3	42.7	29.8	32.3	27.9	34.8±0.33
	Magnitude	2.7e4	36.6	26.5	31.0	26.2	31.3±0.32	7.8e3	39.4	29.9	33.4	27.2	34.0±0.33
	Wanda-sp	1.0e3	38.8	26.1	29.3	26.6	31.4±0.32	4.8e3	38.0	25.8	29.7	26.8	31.3±0.32
	FLAP	5.0e4	37.3	24.7	29.2	25.6	30.5±0.32	5.3e3	38.0	26.3	29.4	26.8	31.3±0.32
	LLM-Pruner	103.3	**48.4**	31.6	43.4	**29.7**	41.0±0.33	5.5e3	41.4	29.7	31.2	28.5	34.0±0.33
	SlimGPT	2.6e4	37.1	27.0	27.8	26.9	30.6±0.32	9.1e3	37.3	25.1	28.6	26.8	30.5±0.32
	SlimLLM	183.5	43.3	30.2	35.5	27.6	36.1±0.33	4.0e3	**45.1**	**37.6**	34.1	30.0	38.3±0.34
	CoCurve	**59.1**	47.1	**33.3**	**44.8**	29.3	**41.4**±0.34	**277.1**	44.3	33.1	**41.8**	**30.0**	**39.5**±0.34
50%	Random	1.3e5	36.1	24.6	29.7	26.6	30.3±0.32	6.5e3	37.6	27.1	28.7	27.2	31.1±0.32
	Magnitude	1.6e5	35.6	26.6	32.4	24.7	31.4±0.32	9.4e3	36.8	26.5	28.1	26.0	30.4±0.32
	Wanda-sp	3.7e5	36.6	22.5	30.4	25.5	30.1±0.31	6.2e3	35.7	25.2	27.6	27.0	29.7±0.32
	FLAP	6.3e5	36.1	24.0	33.3	25.8	31.3±0.31	6.5e3	36.0	26.3	28.5	26.9	30.3±0.32
	LLM-Pruner	865.1	**42.0**	**27.7**	31.3	**27.9**	33.7±0.32	9.3e3	37.5	26.7	29.2	27.0	31.2±0.32
	SlimGPT	1.1e5	36.7	25.7	33.5	25.3	31.9±0.32	6.1e3	36.1	24.6	29.8	25.9	30.3±0.32
	SlimLLM	**773.3**	39.0	26.0	31.0	26.5	32.0±0.32	9.7e3	36.1	26.1	29.8	26.4	30.7±0.32
	CoCurve	956.4	40.1	27.3	**36.1**	27.5	**34.5**±0.32	**2.3e3**	**38.4**	**27.4**	**33.5**	**28.2**	**33.2**±0.32

Note. Avg_{12} includes stratified SE; bold/underline: first/second; / / show within-block rank, Avg_{12} rank, and /// collapse.

3.2 Language Modeling and Downstream Capabilities

CoCurve preserves quality as compression becomes harder. Across 6 LLMs, CoCurve is best in $53/60$ corpus-level comparisons: $38/45$ on the five-model shared grid and $15/15$ at 70B (Figure 3). The separation grows on distinct architectures: the 8B WikiText advantage rises from 2.6% at 20% to 19.0% and 42.8% at $30/40\%$, while the 24B model reaches 20.5 versus 78.8 perplexity at 30% and 277.1 versus 3299 at 40%. This scale-consistent widening supports interaction-aware selection: as redundancy shrinks, compatibility among jointly pruned units becomes more consequential. CoCurve therefore advances the quality–deployment frontier: at 30%, it wins 94% of 18 model–corpus comparisons; on the five-model $30/40\%$ grid, it permits 2.2–6.6 points more pruning at matched quality in $9/10$ cases (Appendix D.1.4).

The advantage transfers from likelihood to downstream decisions. Across all eight 8B/24B model–ratio blocks from 20–50%, CoCurve leads Avg_{12} by 0.4–2.0 points over the strongest baseline and ranks first or second in $31/32$ capability-group comparisons (Table 1). At the 50% boundary on the 8B model, it remains first or second on eight of twelve individual tasks. It also leads all 12 dense-agreement group comparisons across 20–40% (Appendix D.5.1), linking lower predictive risk to preserved decisions rather than an isolated perplexity gain.

The advantage persists with in-context demonstrations. Across five model–ratio blocks, CoCurve leads every subject-matched five-shot MMLU comparison by 0.2–2.1 points (Table 23). The ordering persists when demonstrations change the inference context, indicating that the selected structures preserve usable knowledge beyond zero-shot behavior.

Inherited capacity yields a better memory–capability point. At 12.14 GB peak memory, the recovered and quantized 24B checkpoint uses 14% less memory than dense bf16 Mistral-7B yet improves all four capability groups and raises Avg_{12} by 1.7 points. The result directly shows that pruning can retain useful capacity from a larger pretrained parent, yielding higher quality than a native smaller checkpoint under a tighter memory budget (Appendix D.6).

Table 2: **Pruned large versus native small.**

Checkpoint	Prec.	Peak↓	Cmn_4↑	Sci_3↑	Read_4↑	Know_1↑	Avg_{12}↑
Mistral-Small-24B versus Mistral-7B							
Mistral-Small-24B dense	bf16	45.02	73.5	63.4	65.9	52.3	66.7
Mistral-7B dense	bf16	14.16	69.5	57.4	61.9	44.1	61.8
Mistral-Small-24B (20%)	NF4	**12.14**	**69.6**	**59.5**	**64.9**	**45.5**	**63.5**

Note. Peak is forward-workload GB. Gray marks the dense parent; blue marks the favorable matched point.

3.3 Recovery and Quantization

Structure sets the recovery frontier. A useful structure should remain recoverable under a common lightweight adaptation. We apply one rank-16 recipe to five masks on an 8B Llama and a 24B Mistral, using a single lightweight pass over cleaned Alpaca for every mask. The 8B training stage takes only 1.2–1.9 GPU-hours per mask, excluding evaluation. Across all eight blocks, CoCurve ranks top-two in 14/16 WikiText-2/Avg_{12} columns and leads both 8B metrics from 30–50% (Table 3). Across six scales, the recipe lowers perplexity at all 24 measured points; at 50%, it raises 8B Avg_{12} by 10.8 points and cuts selection-only perplexity from 41.7–4616.9 to 12.6–38.0 (Figure 3). Thus CoCurve preserves an adaptable subnetwork, not merely a favorable unrecovered checkpoint.

Table 3: **Common-LoRA recovery across model scales.** Full results: Appendix D.2.1.

	20%		30%		40%		50%	
Method	**Wiki↓**	**Avg_{12}↑**	**Wiki↓**	**Avg_{12}↑**	**Wiki↓**	**Avg_{12}↑**	**Wiki↓**	**Avg_{12}↑**
Llama-3.1-8B-Instruct	**Dense:** Wiki 6.4; Avg_{12} 65.2							
Random	11.3	56.8	15.3	51.9	20.0	47.5	32.1	43.2
LLM-Pruner	**9.9**	56.0	13.0	51.3	17.4	47.9	28.0	44.6
SlimGPT	10.9	55.6	14.7	51.8	24.3	45.3	41.8	40.7
SlimLLM	10.2	56.8	13.9	51.7	19.6	47.4	31.1	42.0
No LoRA	12.9	56.0	21.6	49.0	59.1	41.4	956.4	34.5
CoCurve	10.0	**57.1**	**12.6**	**53.7**	**16.3**	**49.6**	**24.2**	**45.3**
Mistral-Small-24B	**Dense:** Wiki 4.7; Avg_{12} 66.7							
Random	6.7	64.3	8.2	61.0	11.5	57.3	16.4	53.5
LLM-Pruner	6.8	**65.5**	8.3	**62.3**	10.4	58.5	14.4	53.9
SlimGPT	6.7	64.0	8.0	60.5	10.1	55.5	**13.4**	50.3
SlimLLM	**6.4**	65.2	8.5	61.6	11.4	56.9	15.6	51.6
No LoRA	9.0	60.6	20.5	52.9	277.1	39.5	2.3e3	33.2
CoCurve	6.6	64.4	**8.0**	61.7	**9.7**	**58.9**	14.7	**54.3**

Table 4: **Structural pruning and weight quantization on Llama-3.1-8B-Instruct.** Quality and measured deployment cost at the same 30/40% structures; task breakdown: Appendix D.2.2.

			Efficiency				Perplexity ↓			Downstream groups ↑				
ρ	**Method**	**Prec.**	**Param.↓**	**Mem.↓**	**Pref.↑**	**Lat.↓**	**Wiki**	**PTB**	**C4**	**Cmn_4**	**Sci_3**	**Read_4**	**Know_1**	**Avg_{12} ±SE**
–	**Dense**	bf16	8.03	15.99	4112	31.74	6.4	10.4	10.7	70.9	60.1	67.6	47.8	65.2±0.34
		INT8	8.03	9.50	3035	133.06	6.5	10.4	10.9	70.9	60.2	67.4	47.7	65.1±0.34
		NF4	8.03	6.34	3934	47.90	6.8	10.8	11.5	70.7	58.4	66.8	46.7	64.3±0.34
30%	LLM-Pruner	INT8	5.93	7.54	3659	132.76	26.8	39.6	40.8	54.6	38.1	**52.1**	31.8	47.8±0.34
		NF4	5.93	5.34	5065	44.65	29.2	42.2	44.3	**54.4**	37.2	**51.1**	31.4	47.1±0.34
	SlimLLM	INT8	5.93	7.54	3726	130.36	80.0	102.1	115.3	48.6	36.5	41.0	30.5	41.5±0.34
		NF4	5.93	5.34	5266	44.10	80.0	101.5	114.4	48.2	36.7	41.3	30.4	41.6±0.34
		bf16	5.93	12.08	5232	30.64	21.6	36.3	32.0	55.7	41.7	51.8	33.4	49.0±0.35
	CoCurve	INT8	**5.93**	**7.54**	**3606**	134.61	**21.7**	**36.9**	**32.3**	**55.6**	**41.7**	51.7	**33.3**	**48.9**±0.35
		NF4	**5.93**	**5.34**	**5013**	**46.14**	**23.1**	**38.5**	**34.8**	54.1	**38.4**	50.9	**32.4**	**47.3**±0.34
40%	LLM-Pruner	INT8	5.23	6.89	3931	130.22	101.3	162.9	124.9	**48.4**	32.1	43.7	**29.6**	**41.2**±0.34
		NF4	5.23	5.00	5464	44.56	116.1	194.2	138.8	**48.3**	30.8	42.2	**29.1**	**40.3**±0.33
	SlimLLM	INT8	5.23	6.89	4025	130.34	187.8	219.6	222.5	42.9	29.5	36.4	27.7	36.1±0.33
		NF4	5.23	5.00	5942	42.90	192.8	272.0	242.6	43.6	29.9	35.4	27.7	36.1±0.33
		bf16	5.24	10.79	5654	31.45	59.1	108.4	76.1	47.1	33.3	44.8	29.3	41.4±0.34
	CoCurve	INT8	**5.24**	**6.89**	**3889**	**131.11**	**57.6**	**105.4**	**75.0**	46.2	**33.0**	**44.7**	29.2	41.0±0.34
		NF4	**5.24**	**5.00**	**5413**	**44.87**	**66.9**	**140.8**	**86.7**	45.8	**31.3**	**43.8**	28.6	40.1±0.34

Note. Params: B; peak: GB; prefill: tokens/s; latency: ms/step. Blue marks CoCurve gains over matched-precision Dense; shading ranks quality within (ρ, precision).

Low-bit weights preserve the selection advantage. We apply INT8 and NF4 to fixed masks from three distinct pruners, so within-precision comparisons isolate the selected structure. Across both ratios and precisions, CoCurve leads all 12 corpus-level perplexity comparisons and is first or within 0.2 points of first on every Avg_{12} column, showing that the selected structure retains its quality advantage across both low-bit formats (Table 4).

Structural and numerical compression compound. Within the same CoCurve structure, NF4 cuts peak memory by 54–56% relative to bf16 while retaining 96% of prefill throughput; Avg_{12}

changes by only 1.3–1.7 points. Relative to dense NF4, physically slicing the $30/40\%$ structures further lowers memory by $16/21\%$ and raises prefill throughput by $27/38\%$. Thus low-bit weights compress numerical storage, while CoCurve removes dense computation itself; their composition captures both gains at one deployment point.

3.4 Inference efficiency

Table 5: **Inference gains from physically sliced checkpoints.** Absolute measurements and matched-dense gains.

Model	ρ	Param.↓	Prefill↑	Decode↑	Peak↓
Scale at a fixed 20% pruning budget					
Llama-3.2-3B	20%	2.64	10027 (1.17×)	548 (0.95×)	6.11 (0.83×)
Falcon3-7B	20%	6.12	6059 (1.18×)	561 (1.03×)	13.59 (0.84×)
Llama-3.1-8B-It	20%	6.63	5200 (1.21×)	503 (1.01×)	14.29 (0.83×)
Mistral-Nemo-12B	20%	10.06	3485 (1.20×)	404 (1.09×)	21.36 (0.83×)
Compression levels on Llama-3.1-8B-Instruct					
Llama-3.1-8B-It	30%	5.93	5524 (1.28×)	504 (1.02×)	12.94 (0.75×)
Llama-3.1-8B-It	40%	5.24	6098 (1.41×)	512 (1.04×)	11.53 (0.67×)
Llama-3.1-8B-It	50%	4.53	6801 (1.58×)	511 (1.06×)	10.10 (0.59×)

Note. bf16, eager attention. Params: B; prefill/decode: tokens/s; generation peak: GB. Blue numbers in parentheses are ratios to dense; ρ is the removable-projection budget.

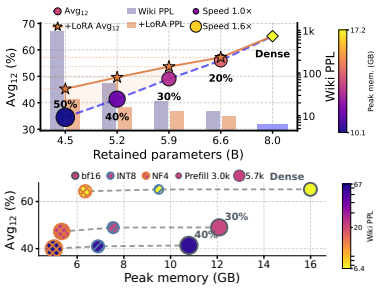

Figure 4: **Quality–efficiency frontiers.** Cross-model view: Appendix D.3.

Physical slicing realizes deployment gains across scale and compression. At 20% from 3B to 12B, CoCurve raises prefill throughput by 17–21%, lowers peak memory by 16–17%, and keeps decode within 5% of dense or improves it by up to 9%. Along the 8B path, prefill rises from $1.28\times$ to $1.58\times$ and peak memory falls from $0.75\times$ to $0.59\times$ dense, while decode remains at 1.02–$1.06\times$ (Table 5). The masks therefore compile into smaller dense workloads rather than logical sparsity.

Selection, recovery, and precision move complementary frontier axes. At matched $20/40\%$ budgets, CoCurve shares the narrow throughput–memory band of other sliced selectors yet retains higher quality (Table 18). In Figure 4, LoRA moves each unchanged checkpoint toward dense quality, whereas NF4 moves it toward lower memory. One graph therefore supports independently controlled structure, recovered quality, and numerical precision.

3.5 Transfer to vision–language models

Table 6: **VLM transfer/recovery.** Full: Table 20.

Model	Method	20% Sel.	20% +LoRA	30% Sel.	30% +LoRA	40% +LoRA	50% +LoRA
Qwen2.5-VL-7B	**Dense Avg_7: 81.6**						
	Magnitude	72.9	75.1	58.7	69.2	61.7	48.3
	Wanda-sp	62.9	69.2	52.2	61.4	50.3	35.8
	FLAP	73.3	75.3	58.5	68.7	61.5	42.6
	LLM-Pruner	72.2	75.1	57.2	**70.9**	61.4	50.1
	CoCurve	**74.4**	**76.2**	**63.6**	69.7	**61.9**	**55.1**
Qwen3-VL-8B	**Dense Avg_7: 83.4**						
	Magnitude	57.0	68.8	37.7	58.3	35.6	33.3
	Wanda-sp	40.5	51.3	35.9	40.8	35.6	34.0
	FLAP	68.4	70.7	46.5	63.9	54.5	34.0
	LLM-Pruner	65.8	72.8	47.0	68.6	**59.8**	**36.6**
	CoCurve	**71.1**	**75.1**	**57.7**	**69.5**	52.1	34.8
InternVL3-8B	**Dense Avg_7: 83.6**						
	Magnitude	70.7	75.0	45.2	64.6	56.4	33.8
	Wanda-sp	49.8	51.0	45.0	49.6	44.9	35.2
	FLAP	74.8	76.6	61.3	**69.9**	54.6	33.8
	LLM-Pruner	71.8	75.8	60.3	68.7	55.1	41.7
	CoCurve	**75.9**	**76.9**	**62.4**	69.2	**59.5**	**53.8**

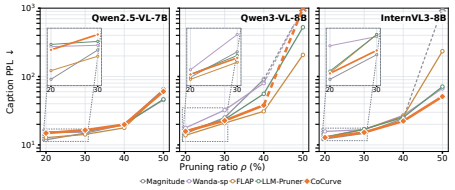

Figure 5: **Recovered caption perplexity ↓.**

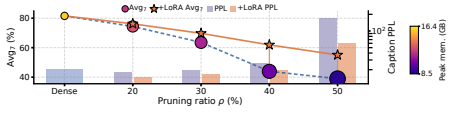

Figure 6: **Qwen2.5-VL quality–efficiency.** Full: Appendix D.4.3.

Co-pruning transfers without tower-specific design. One teacher-KL graph and conditional pruner jointly score and prune units across the vision and language towers, without separate objectives or manually assigned tower budgets. CoCurve leads all 6 pruning-only Avg_7 blocks across the three VLMs at $20/30\%$. At 30%, it improves over the strongest external pruner by 4.9, 10.7, and 1.1 points; removing only the edge term loses 2.5–7.6 points under identical features, guards, and budgets. The consistent separation at this harder point shows that one graph-conditioned objective transfers across VLM architectures without bespoke tower allocations.

The selected multimodal structures remain recoverable. Under one LoRA recipe across methods, models, and ratios, CoCurve ranks first or second in $11/12$ Avg_7 blocks through 50% pruning and

leads both Qwen2.5-VL-7B and InternVL3-8B at that boundary (Table 6). The caption-perplexity curves expose the same recovery boundary without changing masks: edges select a viable structure, and lightweight adaptation recovers capability within it (Figure 5). After slicing, Qwen2.5-VL reaches $1.23\times$ dense prefill throughput with 43% lower peak memory at 50%; memory falls monotonically across all three VLMs (Figure 6).

3.6 Interaction-aware selection and calibration

Table 7: **Contribution of co-pruning edges.** Full results: Table 30.

ρ	Selection rule	Δ KL ↓	Δ Wiki ↓	Δ Avg$_{12}$ ↑
20%	Diagonal	0.000	0.0	0.0
	Edges, fixed split	**-0.049**	-0.4	+1.2
	CoCurve	-0.032	**-0.3**	**+2.0**
30%	Diagonal	0.000	0.0	0.0
	Edges, fixed split	**-0.240**	-8.2	+1.1
	CoCurve	-0.108	**-9.3**	**+1.4**

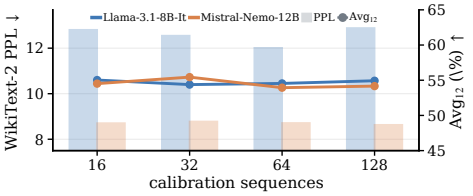

Figure 7: **Calibration-data scaling.** Full sensitivity: Appendix E.4.

Co-pruning edges improve decisions beyond diagonal saliency. Holding the diagonal selector's attention/FFN allocation fixed, edge-conditioned reordering lowers independent test KL by $0.049/0.240$ and WikiText-2 perplexity by $0.4/8.2$ at $20/30\%$, isolating the interaction benefit. Joint competition then gives the largest Avg$_{12}$ gains ($2.0/1.4$ points) and the largest WikiText-2 reduction at 30% (9.3). The fixed-split arm has lower test KL in all four blocks but joint allocation has higher Avg$_{12}$ in three, reflecting that label-free next-token KL preserves predictive fidelity rather than exactly ranking a heterogeneous task mixture. Against diagonal selection, CoCurve nevertheless improves test KL, perplexity, and Avg$_{12}$ in every block without downstream labels.

The graph needs only modest calibration data. Rebuilding $\mathbf{H}$ and reselecting from 16–128 sequences changes WikiText-2 PPL by at most 0.87 and Avg$_{12}$ by at most 1.50 points across both models (Figure 7). Across 26 interaction paths, every path places at least 98.5% of held-out resamples within one standard error of minimum risk (Appendix E.4), showing that both graph construction and held-out selection are stable.

4 What the Co-Pruning Edges Reveal

Having established a consistent quality advantage, we next ask what the co-pruning graph captures and when it matters. The evidence forms one chain: edges organize across modules, expose units missed by isolated saliency, reverse a conditional decision, and predict disproportionate damage under matched removal. We therefore assess effective interaction strength from changed selections and behavior, rather than from the numerical path coordinate alone.

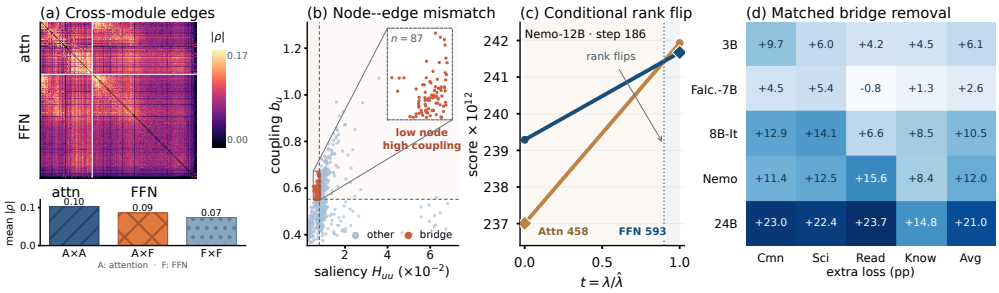

Figure 8: **Co-pruning edges connect structure, decisions, and behavior.** Full analyses: Appendix E.1.

Interactions span attention and FFN. For pairs with positive diagonals, normalize each off-diagonal as $|\gamma_{uv}| = |\mathbf{H}_{uv}|/\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}$, excluding self-correlations and zero-diagonal rows. Across 5 language models, mean attention–FFN magnitude is 0.74–0.98 times the within-module mean

(Figure 8a), with the same organization across 10 LLMs and 7 VLMs (Appendix E.1). Shared-budget allocation can change module totals; pair-specific cross-module structure becomes actionable only when each candidate's cost depends on the units already removed.

Low saliency can coexist with strong coupling. Set $d_u = \mathbf{H}_{uu}$ and $b_u = \sum_{v \neq u} |\mathbf{H}_{uv}|$; bridge units lie below the 40th percentile of d_u but above the 60th percentile of b_u. This mask-independent intersection contains $87/768$ units on Llama-3.1-8B-Instruct and 3.5–14.5% across 10 LLM matrices (Figure 8b), exposing a blind spot of diagonal saliency that CoCurve addresses by conditioning on the current removal set.

Accumulated edges reverse an actual decision. Figure 8c compares eligible units at one greedy prefix. Attention unit 458 has lower node cost than FFN unit 593 but a larger accumulated edge term, so their scores cross at $0.90\hat{\lambda}$ and reverse the node-only ranking. The diagonal mask removes units 458 and 472, whereas CoCurve retains both and removes 593. This trace records an actual conditional decision: Table 25 gives its components, Corollary 6 generalizes the crossing, and Proposition 8 bounds its stability under estimation error.

The bridge population matters beyond its individual scores. On 5 models, we force-remove bridges or equal-size low-coupling sets matched unit-for-unit on saliency, type, and parameter cost; a shared greedy fill fixes total cost and the attention/FFN split. Bridge removal degrades $19/20$ capability groups by up to 23.7 points, including 14.8–23.7 across all four Mistral-Small-24B groups (Figure 8d). Thus the matrix identifies low-saliency units whose joint removal is disproportionately damaging. A stricter layer-matched intervention shows the same direction on both models, raising test KL and reducing Avg_{12} (Table 27).

5 Related Work

Structured LLM pruning. Existing pruners score heads, channels, or blocks using gradient signals in LLM-Pruner (Ma et al., 2023), activation statistics in FLAP and SlimLLM (An et al., 2024; Guo et al., 2025), or OBS reconstruction in SlimGPT (Ling et al., 2024). They may recover a fixed structure with LoRA (Ma et al., 2023; Ling et al., 2024), interleave adaptation and pruning in LoRAPrune (Zhang et al., 2024), or optimize the retained basis and allocation (Ashkboos et al., 2024; Li et al., 2025; Wang et al., 2026). CoCurve instead makes the compatibility of *joint removals* the selection signal and lets attention heads and FFN groups compete within one conditional objective.

Curvature and removal interactions. OBD established diagonal curvature saliency (LeCun et al., 1989), while OBS introduced curvature-aware compensation (Hassibi & Stork, 1992). Later work models weight-space or structured interactions for joint selection and rearrangement (Yu et al., 2022; Nonnenmacher et al., 2022; Kwon et al., 2022), or combines curvature with repair (van der Ouderaa et al., 2024). SAViT coordinates vision-Transformer structures with cross-component Taylor terms (Zheng et al., 2022). Our distinction is an output-space, cross-layer attention–FFN graph estimated from finite forward ablations and reused for conditional pruning at multiple budgets.

Forward perturbations and multimodal compression. Bonsai learns an additive predictor from forward removals (Kolawole et al., 2024); CoCurve's geometry additionally exposes pairwise effects without enumerating pairs. Efficient-VLM work mostly prunes visual tokens (Chen et al., 2024; Zhang et al., 2025; Cao et al., 2024; Zou et al., 2026), while parameter-structural methods use component-specific search and recovery (Shi et al., 2023; Liang et al., 2025). We instead test one predictive-risk graph across both towers; Appendix F.1 gives the full taxonomy.

6 Conclusion

We presented CoCurve, reframing structured pruning from ranking isolated units to selecting a jointly removable set. CoCurve estimates set-dependent predictive risk from forward responses and uses signed co-pruning interactions to condition pruning decisions and compile smaller dense models. Across architectures and deployment conditions, this set-level view advances the quality–deployment frontier and traces the gain to joint-removal interactions rather than budget allocation alone. More broadly, compression should ask not only which components matter in isolation, but which can be removed together while preserving the dependencies that sustain capability, offering a path toward smaller, more accessible foundation models.

Reproducibility statement. The method, structured-unit semantics, complete algorithms, and complexity are specified in Appendix A; complete proofs appear in Appendix B; and Appendix C records models, data, baselines, seeds, and evaluation conventions. Code is included in the supplementary material, together with experiment configurations, the evaluation harness, and scripts that regenerate all reported tables from structured evaluation outputs. Appendix F records compute and asset details.

Ethics statement. This work studies compression of publicly released pretrained models using public benchmark and calibration data. It uses no human subjects, private data, or personally identifying information. Licenses, limitations, and broader-impact considerations are documented in Appendix F.

AI use statement. Generative AI tools were used to assist with language editing, presentation and organization, literature search, and feedback on the communication of experimental results. The research questions, methodology, theoretical analysis, implementation, experiments, and scientific conclusions were developed and verified by the authors. All AI-assisted material was reviewed and revised by the authors, who take full responsibility for the paper.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 10865–10873, 2024. doi: 10.1609/aaai.v38i10.28960.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. MADTP: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 *Computer Vision – ECCV 2024*, 2024.

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

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Falcon-LLM Team. The falcon 3 family of open models. https://huggingface.co/blog/falcon3, 2024.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhiren Gong, Yikun Hou, Fan Wu, Che Wang, Fuyao Zhang, Tiantong Wu, Yurong Hao, Jiaming Zhang, Yiyang Duan, Tiantong Wang, Fei Huang, Chau Yuen, and Wei Yang Bryan Lim. SubspacePath pruner: Inference-time pruning via probe-based representation–parameter coupling. In *International Conference on Machine Learning (ICML)*, 2026a. URL https://openreview.net/forum?id=fvkCjFvWKf.

Zhiren Gong, He Lu, Tiantong Wang, Yichi Zhang, Yixin Wang, Zihao Zeng, Ming Xiao, Chau Yuen, and Wei Yang Bryan Lim. Conditional co-ablation: Recovering self-repair backups in transformer circuits. *arXiv preprint arXiv:2607.01940*, 2026b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. SlimLLM: Accurate structured pruning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20766–20776. PMLR, 2025.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html.

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper/1992/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Steven Kolawole, Lucio Dery, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. Everybody prune now: Structured pruning of LLMs with only forward passes. *arXiv preprint arXiv:2402.05406*, 2024.

Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper/1989/hash/6c9882bbac1c7093bd25041881277658-Abstract.html.

Guanchen Li, Yixing Xu, Zeping Li, Ji Liu, Xuanwu Yin, Dong Li, and Emad Barsoum. Týr-the-Pruner: Structural pruning LLMs via global sparsity distribution optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.52202/085713-4989.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. EfficientLLaVA: Generalizable auto-pruning for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9445–9454, 2025.

Gui Ling, Ziyang Wang, Yuliang Yan, and Qingwen Liu. SlimGPT: Layer-wise structured pruning for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/, 2024b.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.1035.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Mistral AI Team. Mistral-NeMo-base-2407 model card. `https://huggingface.co/mistralai/Mistral-Nemo-Base-2407`, 2024. Accessed 2026-08-31.

Mistral AI Team. Mistral-small-24b-base-2501 model card. `https://huggingface.co/mistralai/Mistral-Small-24B-Base-2501`, 2025. Accessed 2026-08-31.

Leandro Giusti Mugnaini, Bruno Lopes Yamamoto, Lucas Lauton de Alcantara, Victor Zacarias, Edson Bollis, Lucas Pellicer, Anna Helena Reali Costa, and Artur Jordao. Efficient LLMs with AMP: Attention heads and MLP pruning. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2025. doi: 10.1109/IJCNN64981.2025.11227985. URL `https://doi.org/10.1109/IJCNN64981.2025.11227985`.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Manuel Nonnenmacher, Thomas Pfeil, Ingo Steinwart, and David Reeb. SOSP: Efficiently capturing global correlations by second-order structured pruning. In *International Conference on Learning Representations (ICLR)*, 2022.

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 (JMLR)*, 21(140):1–67, 2020.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8722–8731, 2020. doi: 10.1609/aaai.v34i05.6398.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Fabrizio Sandri, Elia Cunegatti, and Giovanni Iacca. 2SSP: A two-stage framework for structured pruning of LLMs. *Transactions on Machine Learning Research*, 2025.

Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. UPop: Unified and progressive pruning for compressing vision-language transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

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

Sidak Pal Singh and Dan Alistarh. WoodFisher: Efficient second-order approximation for neural network compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4149–4158, 2019.

Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. Faster gaze prediction with dense networks and fisher pruning. *arXiv preprint arXiv:1801.05787*, 2018.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tycho F. A. van der Ouderaa, Markus Nagel, Mart van Baalen, and Tijmen Blankevoort. The LLM surgeon. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Ziyang Wang, Jiangfeng Xiao, Chuan Xiao, Ruoxiang Li, Rui Mao, and Jianbin Qin. GRASPrune: Global gating for budgeted structured pruning of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10719–10736, 2026. doi: 10.18653/v1/2026.acl-long.491.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Alex Young, Bei Chen, Chao Li, et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2:67–78, 2014.

Xin Yu, Thiago Serra, Srikumar Ramalingam, and Shandian Zhe. The combinatorial brain surgeon: Pruning weights that cancel one another in neural networks. In *International Conference on Machine Learning (ICML)*, 2022.

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

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3013–3026. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.178. URL https://aclanthology.org/2024.findings-acl.178/.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025.

Chuanyang Zheng, Zheyang Li, Kai Zhang, Zhi Yang, Wenming Tan, Jun Xiao, Ye Ren, and Shiliang Pu. SAViT: Structure-aware vision transformer pruning via collaborative optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Yulin Zou, Wenyan Chen, Yan Chen, Anya Rajan, JooYoung Park, Shivaraman Nitin, Luo Tao, Francisco Romero, and Dmitrii Ustiugov. Codecsight: Leveraging video codec signals for efficient streaming VLM inference. *arXiv preprint arXiv:2604.06036*, 2026.

APPENDIX OVERVIEW

This appendix supplies the method specification, theoretical foundations, complete experimental record, mechanism analysis, and reproducibility material for CoCurve. To keep that record navigable, it is organized into six parts that follow the paper's argument.

Part A: Method Details. This part turns the main method into a complete implementation path.

- Appendix A.1: notation and the common attention–FFN unit inventory.
- Appendix A.2: single-unit perturbations and blockwise Fisher–Gram construction.
- Appendix A.3: conditional shared-budget selection, complexity, and breakpoint-guided interaction selection.

Part B: Theoretical Foundations. This part connects the objective, estimator, and solver rather than treating them as independent results.

- Appendix B.1: local teacher-KL geometry and its structured curvature.
- Appendix B.2: the exact Gram identity, finite-ablation surrogate, and approximation remainder.
- Appendix B.3: edge-induced ranking reversals, feasibility, the finite breakpoint path, and conditional-decision stability.
- Appendix B.4: the diagonal Fisher limit, the geometry of normalized edges, and the co-pruning mass omitted by isolated scores.

Part C: Experimental Setup and Implementation Details. This part fixes the protocol before the complete results are shown.

- Appendix C.1: datasets, task groups, metrics, and aggregation.
- Appendix C.2: models, pruning budgets, and shared guards.
- Appendix C.3: baseline definitions and the matched comparison protocol.
- Appendix C.4: language-model calibration, metrics, and reporting conventions.
- Appendix C.5: vision–language calibration and transfer protocol.
- Appendix C.6: common LoRA, quantization, and physical-slicing configurations.

Part D: Complete Compression Results. This part mirrors the order of the main empirical section.

- Appendix D.1: full LLM quality and the measured deployment-frontier shift.
- Appendix D.2: common-LoRA recovery and weight quantization.
- Appendix D.3: physical inference efficiency and selection-cost accounting.
- Appendix D.4: full VLM transfer, diagonal control, and recovery.
- Appendix D.5: behavioral fidelity and few-shot prompting.
- Appendix D.6: pruning a larger model versus starting from a native smaller checkpoint.

Part E: What the Co-Pruning Edges Reveal. This part expands the main paper's explanatory evidence as one mechanism chain.

- Appendix E.1: graph structure, conditional decision reversals, bridge interventions, and learned allocation.
- Appendix E.2: module eligibility and interaction-granularity controls.
- Appendix E.3: same-budget discrimination by the quadratic surrogate.
- Appendix E.4: split-half, calibration-budget, and held-out-selection stability.

Part F: Discussion and Reproducibility. This part closes the paper with neighboring formulations, an evidence-matched scope statement, and release details.

- Appendix F.1: structured pruning, curvature, and forward perturbations.
- Appendix F.2: demonstrated scope and natural extensions.
- Appendix F.3: broader impact.
- Appendix F.4: assets, licenses, and ethics.
- Appendix F.5: reproducibility commitments.

A METHOD DETAILS

This part gives a self-contained account of the method in implementation order. Attention heads and feed-forward network (FFN) groups write to the model's shared residual stream (Vaswani et al., 2017; Elhage et al., 2021), so pruning one can alter how another affects later predictions (McGrath et al., 2023; Gong et al., 2026b). CoCurve therefore places all such units in one joint inventory: teacher Kullback–Leibler (KL) risk defines node weights for individual risk and signed co-pruning edges, single-unit ablations estimate both, conditional pruning constructs removal sets, and held-out risk selects the interaction level. After fixing units and gate semantics in Appendix A.1, Appendix A.2 constructs the graph and Appendix A.3 turns it into a physically pruned checkpoint.

A.1 NOTATION AND STRUCTURED UNITS

Table 8 collects only the objects that recur across the method and theory. Local loop indices and one-off implementation quantities are defined where they are used.

Table 8: **Core notation.**

Symbol	Meaning
	Units and budget
$\mathcal{U}$	Joint inventory $\{u_1, \dots, u_M\}$ of removable attention units and FFN groups
s, S	Gate vector $s = (s_u)$ (continuous near $\mathbf{0}$, binary at deployment) and set $S = \{u : s_u{=}1\}$
c_u, ρ	Removable parameter cost and target fraction relative to $\sum_u c_u$
	Predictive risk and graph
f_0, p_0, p_s	Frozen dense model and dense/gated next-token distributions
$\mathcal{C}, \mathcal{C}'$	Disjoint graph-estimation and held-out model-selection splits
$\mathcal{R}(s)$	Expected token-level teacher KL, $\mathbb{E}_{x,t}[D_{\mathrm{KL}}(p_0 \Vert p_s)]$
$\mathbf{H}^\star, \mathbf{H}$	Exact mask Hessian and finite-ablation Gram estimate
$\mathbf{H}_{uu}, \mathbf{H}_{uv}$	Diagonal node weight and signed co-pruning edge weight
$\delta z_u, \bar{D}_u$	Single-unit next-token-logit response and its stacked Fisher feature
	Conditional selection
$\lambda, \hat{\lambda}, \mathcal{F}$	Interaction-path coordinate, its selected representative, and the candidate removal family
$\widehat{\mathcal{R}}_\lambda$	Quadratic risk with off-diagonal edges weighted by λ
$g_u^{(S)}, \Delta\widehat{\mathcal{R}}_\lambda(u \mid S)$	Incident-edge sum and incremental surrogate risk

Mask semantics and physical pruning. Attention units. For multi-head attention (MHA), one head is one unit; pruning head i removes the corresponding rows of W_Q, W_K, W_V and the input columns of W_O. For grouped-query attention (GQA) (Ainslie et al., 2023) with $n_g = n_q/n_{kv}$ query heads per key/value (KV) head, the unit is the KV-aligned query-head group $\mathcal{I}_g = \{gn_g, \dots, (g{+}1)n_g{-}1\}$; pruning group g removes the rows of W_Q for $i \in \mathcal{I}_g$, the rows of W_K, W_V for KV head g, and the matching input columns of W_O. The head dimension d_h is preserved during slicing; the attention module stores the reduced n'_q, n'_{kv} explicitly.

FFN groups. The intermediate dimension is partitioned into n_f contiguous groups. The FFN output is linear in the post-activation intermediate vector and therefore decomposes exactly into the sum of these groups' contributions. At runtime, pruning group j zeros that post-activation slice before the down projection; physical pruning removes the corresponding rows of $W_{\text{gate}}, W_{\text{up}}$ (or W_1) and columns of W_{down} (or W_2). The graph construction and solver require only this deployable unit registry, so changing n_f changes the deployment granularity and graph size but not the objective or algorithm.

Mask boundary and equivalence. A runtime mask zeros only a unit's residual-stream contribution and never alters normalization, rotary position embedding (RoPE), residual addition, causal/padding masks, GQA repeat rules, or dropout (the model is in eval mode). Evaluation and timing require two distinct checks. Zeroing a unit's weight slices is bit-identical to masking it at runtime, which we verify on a held-out batch, so a masked evaluation and a zeroed one are the same measurement.

Slicing the weight matrices instead is what the timings run on, and its correctness is checked by the realized parameter count of every timed model: on Llama-3.1-8B-Instruct at ρ=20% the sliced model holds 6.633B parameters against the dense 8.030B, and the logit equivalence of the slicing path is verified on a synthetic grouped-query Llama to 10^{-4}. Reported speedups therefore come from a smaller model, not from zeroed weights in a full-size one.

A.2 Forward Estimation

Algorithm 1 turns each unit's next-token-logit response into one column of the Fisher feature matrix. The crux (Proposition 3) is that gathering every ablation on the same teacher top-r support $\mathcal{V}_r(x,t)$ places all responses in one coordinate system. Their Gram matrix then yields every node weight and signed edge without model interventions beyond the M single-unit sweeps. Positive inner products mean reinforcing Fisher-weighted logit responses; negative ones mean partial cancellation. We never materialize the full vocabulary-by-vocabulary Fisher matrix.

Algorithm 1 CoCurve edge estimation (single-unit, top-r Fisher Gram)

Require: frozen f_0, calibration $\mathcal{C}$, units $\mathcal{U}$, top-r, positions per sequence P_{tok}
1: **Teacher pass:** for each $x \in \mathcal{C}$, select token positions $\mathcal{T}(x)$; store top-r indices $\mathcal{V}_r(x,t)$, teacher logits z_0 on that support, and support-conditioned probabilities $p_0^{(r)}$
2: **for** each unit $u \in \mathcal{U}$ **do** $\{M$ calibration sweeps; embarrassingly parallel$\}$
3: run f_{-u} (mask only u); gather z_{-u} on the teacher support $\mathcal{V}_r(x,t)$
4: $\delta z_u \leftarrow z_0 - z_{-u}$; $\widetilde{\delta z}_u \leftarrow \sqrt{p_0^{(r)}} \odot (\delta z_u - \mathbb{E}_{p_0^{(r)}}[\delta z_u])$
5: append $\widetilde{\delta z}_u$ to feature store $\bar{D}_u$ (CPU/memmap)
6: **end for**
7: $\mathbf{H} \leftarrow \frac{1}{P}\bar{D}^\top \bar{D}$ (blockwise Gram); $\mathbf{H} \leftarrow \frac{1}{2}(\mathbf{H} + \mathbf{H}^\top)$
8: assert $\min_u \mathbf{H}_{uu} \geq -\epsilon$ $\{$non-negativity check (Proposition 3)$\}$
9: **return** $\mathbf{H}$

Top-r coordinate consistency. Using a per-unit top-r would place different units in different coordinate systems and invalidate the Gram; the shared teacher support is essential. Defaults: r=256, 128 C4 sequences of length 2048, and deterministic subsampling of P_{tok}=128 positions per sequence, giving $P = 16384$ stored positions. Increasing r from 256 to 512 leaves the selected mask and reported quality unchanged in our support-budget study (Appendix E.4).

A.3 The Greedy Co-Pruning Solver

Algorithm 2 gives the complete pipeline: ablate individual units, form the Gram edge matrix, enumerate candidate removal sets, select one by held-out risk, and physically remove its structures.

Algorithm 2 CoCurve: estimation, held-out selection, and physical removal

Require: frozen f_0, disjoint splits $\mathcal{C}, \mathcal{C}'$, units and costs, ratio ρ, guard
1: $\mathbf{H} \leftarrow$ ESTIMATE$(f_0, \mathcal{C}, \mathcal{U})$ using Algorithm 1
2: $\mathcal{F} \leftarrow$ BREAKPOINTPATH$(\mathbf{H}, \{c_u\}, \rho,$ guard$)$, including the diagonal solution
3: evaluate $\mathcal{R}(S; \mathcal{C}')$ for each distinct $S \in \mathcal{F}$
4: $\widehat{S} \leftarrow \arg\min_{S \in \mathcal{F}} \mathcal{R}(S; \mathcal{C}')$
5: verify the realized budget; slice the weight matrices associated with $\widehat{S}$
6: **return** compressed model $f_{\widehat{S}}$ and its realized removal ratio

Algorithm 3 gives the solver over the joint attention–FFN inventory. For binary gates, $\widehat{\mathcal{R}}_\lambda(s) = \frac{1}{2}\sum_u \mathbf{H}_{uu} s_u + \lambda \sum_{u<v} \mathbf{H}_{uv} s_u s_v$; for removal set S, maintaining $g_u^{(S)} = \sum_{v \in S} \mathbf{H}_{uv}$ makes each step $O(M)$.

Algorithm 3 Cost-normalized greedy co-pruning on a fixed surrogate

Require: $\mathbf{H}$, costs $\{c_u\}$, ratio ρ, interaction weight λ, per-layer cap κ, protected layers $\mathcal{L}_p$
$S \leftarrow \emptyset$; $g \leftarrow \mathbf{0}$; $C_S \leftarrow 0$; $C_\ell \leftarrow 0$ for every layer
$C_\ell^{\mathrm{tot}} \leftarrow \sum_{u:\mathrm{layer}(u)=\ell} c_u$ for every layer
while $C_S < \rho \sum_u c_u$ **do**
  $\mathcal{A} \leftarrow \{u \notin S : \mathrm{layer}(u) \notin \mathcal{L}_p,\ C_{\mathrm{layer}(u)} + c_u \le \kappa\, C^{\mathrm{tot}}_{\mathrm{layer}(u)}\}$; **if** $\mathcal{A} = \emptyset$ **then break**
  $u^\star \leftarrow \arg\min_{u \in \mathcal{A}} \left(\frac{1}{2}\mathbf{H}_{uu} + \lambda\, g_u\right)/c_u$
  $S \leftarrow S \cup \{u^\star\}$; $g \leftarrow g + \mathbf{H}_{:,u^\star}$; $C_S \leftarrow C_S + c_{u^\star}$
  $C_{\mathrm{layer}(u^\star)} \leftarrow C_{\mathrm{layer}(u^\star)} + c_{u^\star}$
end while
return removal set S, $\rho_{\mathrm{actual}} = C_S / \sum_u c_u$, and target-reached flag

Feasibility guards. A per-layer cost cap and first/last-layer protection constrain which units are eligible without altering the conditional score. Both follow one rule—cap $= 1.5\rho$, with the first and last two layers protected—applied identically to every model, ratio, and method, with no per-model choice or reference ratio (Appendix C.2).

A.3.1 Complexity and cost

Estimation is M forward-only ablation passes over $|\mathcal{C}|$ sequences (no backward pass, Hessian-vector product, or pairwise ablation); the Gram is $O(M^2 Pr)$ multiply-adds, accumulated blockwise. The solver is $O(M|S|)$ for $|S|$ units selected for removal. With FFN grouping ($M \sim 10^3$), the Gram and solver remain tractable, and each model needs only one reusable curvature build; measured single-accelerator wall-clock is reported in Appendix D.3.

Why a conditional greedy solver. The guarded, heterogeneous-cost objective in Equation (5) contains minimum knapsack as its diagonal special case and becomes a constrained binary quadratic program when edges are active, so exact subset search is exponential in general. Maintaining the incident sum $g^{(S)}$ makes every greedy step score all eligible additions in $O(M)$ and chooses the exact lowest conditional marginal of the stated surrogate; beam search multiplies this cost by its width, while even one-swap enumeration introduces $O(|S|(M - |S|))$ neighboring sets per iterate. The model-scale audit in Appendix E.3 complements this complexity argument: among 26 same-budget masks, including ten local swaps around the deployed set, the surrogate retains positive local rank correlation with measured risk rather than succeeding only by separating random masks.

A.3.2 Selecting along the interaction path

This section enumerates the finite interaction path, fixes the held-out protocol, and illustrates the selected path points; exact selected interaction strengths for all released masks are recorded in the supplementary artifacts.

Select masks, not surrogate magnitudes. The candidate removal sets, rather than the numerical scale of surrogate risk, determine deployment quality; a common positive rescaling of $\mathbf{H}$ leaves every greedy comparison unchanged. We therefore choose among them using the measured teacher KL of each set.

Breakpoint-guided enumeration. Algorithm 4 gives the numerical path construction. At a fixed greedy prefix, candidate scores are affine in λ. Specifically, the cost-normalized incremental risk is $\mu_u + \lambda\nu_u$, with $\mu_u = \mathbf{H}_{uu}/(2c_u)$ and $\nu_u = g_u^{(S)}/c_u$. A competitor overtakes the current choice only at a crossing. Intersecting these conditions over all steps identifies an interval on which the complete trace is fixed.

The implementation uses float64 scores, smallest-index tie-breaking, a 10^{-12} forward-crossing tolerance, and a 10^{-9} advance beyond each breakpoint. It checks coverage through 1 to within 10^{-9} and raises an error if the solve limit is exhausted. Reported family sizes count the distinct interval sets returned under these numerical conventions; isolated exact-tie solutions and intervals narrower than the advance are not enumerated separately.

Algorithm 4 Numerical breakpoint path for candidate generation

Require: $\mathbf{H}$, costs, ratio, guard; range $[0, 1]$; solve limit $N_{\max} = 400000$
1: $\lambda \leftarrow 0$; $\mathcal{F} \leftarrow \emptyset$; $n \leftarrow 0$
2: **while** $\lambda \leq 1$ and $n < N_{\max}$ **do**
3: run Algorithm 3 at λ, recording prefixes; break score ties by smallest unit index
4: find the earliest finite crossing $\tau > \lambda + 10^{-12}$ after which an eligible competitor becomes strictly better than the chosen unit
5: record removal set S and the next-crossing location; add S to $\mathcal{F}$ if distinct
6: $n \leftarrow n + 1$; **if** $\tau \geq 1$ or no finite crossing exists **then break**
7: $\lambda \leftarrow \tau + 10^{-9}$
8: **end while**
9: **if** the last interval ends below $1 - 10^{-9}$ **then raise** incomplete-path error
10: **return** distinct removal sets $\mathcal{F}$ and recorded intervals

At $\rho = 20\%$, the walk returns 288 masks on Llama-3.2-3B, 303 on Llama-3.1-8B-Instruct, 181 on Falcon3-7B, and 400 on Yi-1.5-9B. A uniform 0.002 grid reaches 199 of the 288 recorded 3B masks and misses the selected minimum. Breakpoint guidance resolves these uneven intervals without choosing a global grid spacing. Enumeration is CPU-only; each distinct candidate requires a held-out masked forward evaluation.

The criterion is stable across held-out splits. λ substantially changes the attention/FFN split—on Llama-3.1-8B-Instruct, attention falls from 38% of removed units at $\lambda = 0$ to 2% at $\lambda = 1$—so we test how strongly the choice depends on the held-out split $\mathcal{C}'$. Over both context lengths and both corpora, at ρ=20% with everything else fixed:

$\mathcal{C}'$ length	$\mathcal{C}'$ corpus	J vs. deployed
1024	C4	0.967
1024	WikiText	0.877
2048	**C4**	*deployed*
2048	WikiText	0.844

The selected removal sets agree with the deployed one at Jaccard 0.84 to 0.97 across the four splits. The selected sets substantially overlap, although they are not identical. We therefore use the deployment context length and C4 split consistently.

Figure 9 shows the path itself on four models. **Path outcomes.** Held-out risk selects an edge-aware removal set when interactions improve prediction and retains the diagonal endpoint otherwise. In the broader ten-model diagnostic, held-out risk rejects the full-strength endpoint on every model, confirming that finite removal requires calibrated rather than undamped local edges. Because the numerical coordinate is not an effect size, Table 29 reports path sizes and the measured consequences of removing or fully undamping the edges. These describe held-out model selection rather than independent statistical significance.

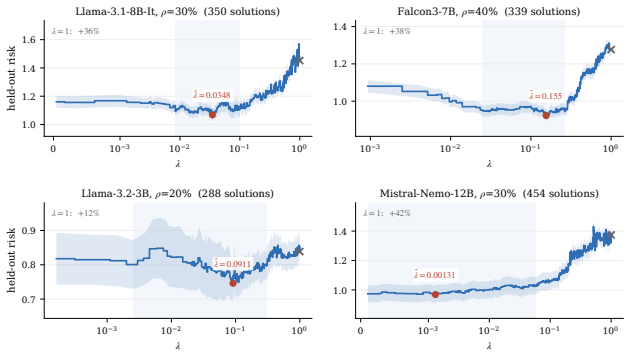

Figure 9: **Held-out selection along the interaction path.** Four held-out risk paths over λ; red dots select $\hat{\lambda}$, bands show one standard error, and crosses mark λ=1. Every minimum is interior, while full-strength interaction is worse.

B THEORETICAL FOUNDATIONS AND GUARANTEES

For M removable units, let $s \in \mathbb{R}^M$ be the smooth removal-gate vector around the dense point $\mathbf{0}$, with $s_u{=}1$ denoting deployment removal. For calibration sequence x and token position t, let z_0, z_s and p_0, p_s be the corresponding dense and gated next-token logits and distributions, write $\Delta z_s = z_0 - z_s$, and define the teacher Kullback–Leibler (KL) risk $\mathcal{R}(s) = \mathbb{E}_{x,t}[D_{\mathrm{KL}}(p_0 \| p_s)]$. This distribution-matching objective uses the dense model as an unlabeled teacher and vanishes at the dense point (Hinton et al., 2015).

The theory follows the method's dependency chain: Appendix B.1 derives its local geometry; Appendix B.2 constructs and diagnoses the finite-ablation Gram estimate; Appendix B.3 shows how its edges change conditional decisions and establishes path, budget, and perturbation properties; and Appendix B.4 identifies the diagonal special case and its omitted interaction term. This chain supports three design claims: individual pruning risk and co-pruning edges arise from the same second-order predictive objective; one forward ablation per unit yields all estimated edges; and their incident sum is the precise surrogate correction that can reverse an isolated ranking. The held-out-risk and fixed-allocation controls in Sec. 3.6, the mechanism interventions in Sec. 4, and the same-budget analysis in Appendix E.3 then connect these mathematical properties to finite-ratio model behavior.

B.1 GEOMETRY OF THE PRUNING RISK

Why the local pruning risk begins at second order.

Lemma 1 (Vanishing zeroth- and first-order terms)**.** *With s ranging over a neighborhood of $\mathbf{0}$, so that $\mathbf{0}$ is interior and $\mathcal{R}$ is differentiable there, $\mathcal{R}(\mathbf{0}) = 0$ and $\nabla_s \mathcal{R}(\mathbf{0}) = 0$; the Taylor expansion at the full model begins at second order.*

Proof. At $s = \mathbf{0}$, $p_{\mathbf{0}} = p_0$, so every term is $D_{\mathrm{KL}}(p_0 \| p_0) = 0$. For the gradient, with $g(s) = D_{\mathrm{KL}}(p_0 \| p_s)$ and p_s a softmax, $\partial_{s_u} g = -\mathbb{E}_{p_0}[\partial_{s_u} z_s] + \mathbb{E}_{p_s}[\partial_{s_u} z_s]$; at $s = \mathbf{0}$, $p_s = p_0$ and the two expectations cancel. Averaging over positions preserves equality. □

The absence of a first-order term makes curvature the first informative carrier of both individual pruning risk and removal interactions.

Proposition 2 (Curvature of teacher-KL pruning risk)**.** *Let the masked logits be twice differentiable near $s = \mathbf{0}$, and let $G = \partial(z_0 - z_s)/\partial s|_{\mathbf{0}}$. For the empirical calibration distribution,*

$$\mathbf{H}^\star = \nabla_s^2 \mathcal{R}(\mathbf{0}) = \mathbb{E}_{x,t}\big[G^\top \big(\operatorname{diag}(p_0) - p_0 p_0^\top\big) G\big]\,. \tag{7}$$

Here $F = \operatorname{diag}(p_0) - p_0 p_0^\top$ at each position, and therefore $\mathbf{H}^\star$ is positive semidefinite (PSD); moreover, $D_{\mathrm{KL}}(p_0 \| p_s) = \frac{1}{2} \Delta z_s^\top F \Delta z_s + O(\|\Delta z_s\|^3)$ locally.

Proof. The gradient of teacher KL with respect to student logits is $p_s - p_0$, which vanishes at $s = \mathbf{0}$, and its logit Hessian there is F. In the second-order chain rule, the second derivatives of the logit map multiply this zero gradient. The remaining term is $(-G)^\top F(-G) = G^\top F G$, the standard Fisher pullback through the gate-to-logit Jacobian (Amari, 1998). Averaging over the finite calibration distribution gives the identity. F is the covariance matrix of a categorical one-hot variable and is positive semidefinite, so every pullback $G^\top F G$ and their expectation are positive semidefinite. Expanding the smooth log-partition function at z_0 gives the stated logit-space remainder. □

For binary gates, expanding $\frac{1}{2} s^\top \mathbf{H}^\star s$ yields diagonal node weights and one signed co-pruning edge per unit pair. CoCurve estimates these derivative columns from single-unit next-token-logit responses on a shared teacher support.

B.2 ESTIMATOR IDENTITIES AND APPROXIMATION DIAGNOSTICS

The full graph from M ablations.

Proposition 3 (Gram equivalence)**.** *Let $\delta z_u = z_0 - z_{-u}$ be the finite next-token-logit response to pruning unit u. For any normalized distribution p on a common support, let $F_p = \operatorname{diag}(p) - pp^\top$*

and $\widetilde{\delta z}_u = \sqrt{p} \odot (\delta z_u - \mathbb{E}_p[\delta z_u])$. *Then* $\delta z_u^\top F_p \delta z_v = \widetilde{\delta z}_u^\top \widetilde{\delta z}_v$. *Consequently, stacking these features over* P *sampled positions as the columns* $\bar{D}_u$ *of* $\bar{D}$ *gives* $\mathbf{H} = \frac{1}{P}\bar{D}^\top \bar{D} \succeq 0$ *and* $\mathbf{H}_{uu} = \frac{1}{P}\|\bar{D}_u\|^2 \geq 0$*, recoverable from* M *single-unit ablations without pairwise model ablations.*

Proof. $a^\top F_p b = \sum_y p_y a_y b_y - (\sum_y p_y a_y)(\sum_y p_y b_y) = \sum_y p_y (a_y - \bar{a})(b_y - \bar{b}) = (\sqrt{p} \odot (a - \bar{a}))^\top (\sqrt{p} \odot (b - \bar{b}))$, where $\bar{a} = \mathbb{E}_p[a]$ and likewise for $\bar{b}$. Apply the identity with $a = \delta z_u, b = \delta z_v$ separately at every sampled position, then stack the resulting features over positions to obtain the Gram, which is PSD with non-negative diagonal. □

Thus $O(M)$ ablations expose $O(M^2)$ edges through inner products, while guaranteeing symmetry, positive semidefiniteness, and a non-negative diagonal in exact arithmetic. Algorithm 1 forms this Gram matrix blockwise, explicitly symmetrizes roundoff, and checks its diagonal; in floating point, positive semidefiniteness holds up to numerical error.

Exact curvature vs. the estimator we use. Two steps are exact; estimating the curvature introduces the approximation.

Remark 4 (What is exact, what is a surrogate). The local expansion $D_{\mathrm{KL}}(p_0 \| p_s) = \frac{1}{2}\Delta z_s^\top F \Delta z_s + o(\|\Delta z_s\|^2)$ (Proposition 2) and the Gram identity $\delta z_u^\top F\, \delta z_v = \widetilde{\delta z}_u^\top \widetilde{\delta z}_v$ (Proposition 3) are exact. We estimate rather than compute the curvature matrix itself. The exact Hessian, which here coincides with the Gauss–Newton matrix because the logit-space gradient vanishes at $s = \mathbf{0}$ (Lemma 1), is $\mathbf{H}^\star = \mathbb{E}[G^\top F G]$, where $G = \partial \Delta z_s / \partial s|_{s=\mathbf{0}} = -\partial z_s / \partial s|_{s=\mathbf{0}}$. We replace each infinitesimal column G_u by the finite single-unit ablation $\delta z_u = z_0 - z_{-u}$ and compute the implemented Fisher inner products on the teacher's top-r support, renormalized to sum to one. The implementation therefore forms a finite-difference, renormalized top-r Fisher–Gram surrogate for $\mathbf{H}^\star$; it equals $\mathbf{H}^\star$ when the logit map is linear in the removal gates and the full vocabulary is retained. At r=256, the median retained teacher mass is 0.993 on Llama-3.2-3B and 0.992 on Mistral-Small-24B, over 16,384 sampled calibration positions per model. Table 34 measures how varying the support affects selection and downstream quality. Its conditional ranking signal is validated directly in Appendix E.3 and Sec. 4.

From local geometry to finite pruning. The exact result is local, whereas the deployed masks remove finite units. Three approximation steps bridge the two: the quadratic model is evaluated away from the origin, derivative columns are replaced by finite ablations, and the Fisher is restricted to a renormalized vocabulary support. The proposition below controls the first step locally; the following measurements test finite-ablation fidelity and joint-logit additivity, while Table 34 tests support sensitivity. Together, these results separate exact local guarantees from measured finite-removal fidelity.

Proposition 5 (Third-order remainder)**.** *Let* $\mathbf{H}^\star = \mathbb{E}[G^\top F G]$ *with* $G = \partial(z_0 - z_s)/\partial s|_{s=\mathbf{0}}$ *be the exact Gauss–Newton curvature of* $\mathcal{R}$*. If the logit map is twice continuously differentiable in the mask variables on a neighborhood of* $s = \mathbf{0}$*, then for every sufficiently small* $\eta > 0$ *whose closed* ℓ_1 *ball lies in that neighborhood, there is a constant* $K_{\mathcal{R}}$*, independent of* s*, with* $\left|\mathcal{R}(s) - \frac{1}{2} s^\top \mathbf{H}^\star s\right| \leq K_{\mathcal{R}} \|s\|_1^3$ *for all* $\|s\|_1 \leq \eta$*.*

Proof. Fix one calibration position and write $f(s) = \Delta z_s$, so $f(\mathbf{0}) = 0$ and $Df(\mathbf{0}) = G$. Bounded second derivatives of the logit map give, uniformly on a sufficiently small ℓ_1 ball, $f(s) = Gs + e(s)$ with $\|e(s)\|_2 \leq L\|s\|_1^2$ and $\|f(s)\|_2 + \|Gs\|_2 \leq K\|s\|_1$ for finite L, K. The softmax log-partition is analytic; its third derivative is bounded on the compact local image of this ball. Taylor's theorem therefore gives $D_{\mathrm{KL}}(p_0 \| p_s) = \frac{1}{2} f(s)^\top F f(s) + r(s)$ with $|r(s)| \leq B\|f(s)\|_2^3$. Moreover,

$$\left|f(s)^\top F f(s) - (Gs)^\top F (Gs)\right| \leq \|F\|_2 \, \|e(s)\|_2 \big(\|f(s)\|_2 + \|Gs\|_2\big) \leq \|F\|_2 L K \|s\|_1^3. \quad (8)$$

Combining the two bounds yields the claim at this position. The empirical calibration distribution is finite, so taking the maximum of its local constants and then averaging establishes one finite constant $K_{\mathcal{R}}$ for $\mathcal{R}$. □

This result explains why a quadratic graph is the right local object; Appendix E.3 asks whether its rankings remain useful for finite removals.

(i) Finite ablation in place of the derivative column. We replace G_u by $\delta z_u = z_0 - z_{-u}$, a secant over the whole mask interval rather than a tangent. Scaling a unit's residual contribution by $1-\epsilon$ and forming $\delta z_u^{(\epsilon)} = (z_0 - z_{u,\epsilon})/\epsilon$ for $\epsilon \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$ lets us compare the two directly. On a sample of 15–18 units per model, stratified by depth, unit type, and saliency, the hard-ablation feature agrees with the finest small-step feature at a median cosine of 0.949–0.981 across four models after the same probability weighting and mean centering as in Proposition 3; the median norm ratio is 1.03–1.07. Thus the logit-response directions and scales used by the estimator remain close to the corresponding small-step proxies. We use fp32 because the small-ϵ difference reaches the bf16 logit noise floor.

(ii) Joint-logit additivity diagnostic. The surrogate uses $\Delta z_S \approx \sum_{u \in S} \delta z_u$. Let $\Delta z_S = z_0 - z_{-S}$ be the true joint displacement and let $\widetilde{\Delta z}_S$ denote its probability-weighted, mean-centered feature, formed as in Proposition 3. We compare the predicted risk $\frac{1}{2} s^\top \mathbf{H} s$ with $\frac{1}{2}\mathbb{E}_{x,t}\|\widetilde{\Delta z}_S\|^2$; this isolates joint-logit additivity from the other approximations.

Across ten nested diagonal-greedy prefixes on Llama-3.2-3B, from two-unit removal through $\rho = 25\%$, the predicted-to-measured quadratic-risk ratio remains within 0.67–1.18 despite extending the local construction to finite removal sets. Both diagnostics rebuild fp32 features on eight sequences of 512 tokens, with held-out text used for the joint-additivity test; the deployed matrix uses bf16 forwards over 128 sequences of 2048.

A uniform positive rescaling of $\mathbf{H}$ leaves selection unchanged (Appendix B.3), so conditional ranking matters more than absolute surrogate scale; held-out predictive risk then chooses among the actual masks, and Appendix E.3 evaluates their ranking fidelity.

B.3 Conditional Pruning and Solver Guarantees

We next establish a non-negative surrogate, identify when an edge changes a decision, and then derive path, budget, and perturbation properties in algorithmic order. Each unit has removable parameter cost $c_u > 0$, and S denotes the current removal set.

Positive-semidefinite path. If $\mathbf{H} \succeq 0$, then $\mathbf{H}_\lambda = (1-\lambda)\,\mathrm{diag}(\mathbf{H}) + \lambda \mathbf{H} \succeq 0$ for every $\lambda \in [0,1]$, and hence $\widehat{\mathcal{R}}_\lambda(s) = \frac{1}{2} s^\top \mathbf{H}_\lambda s \geq 0$ for every real gate vector s. This follows because $\mathrm{diag}(\mathbf{H}) \succeq 0$ and the positive-semidefinite cone is convex.

Conditional marginal and solver cost. Expanding the set objective gives $\Delta\widehat{\mathcal{R}}_\lambda(u \mid S) = \frac{1}{2}\mathbf{H}_{uu} + \lambda g_u^{(S)}$, where $g_u^{(S)} = \sum_{v \in S} \mathbf{H}_{uv}$. The vector $g^{(S)}$ stores these incident-edge sums; updating it by $g \leftarrow g + \mathbf{H}_{:,u^\star}$ costs $O(M)$ per choice and $O(M|S|)$ for a complete fixed-λ solve.

Corollary 6 (Edge-induced ranking reversal)**.** *Fix a greedy prefix S and two eligible units u, v with positive costs. Let $\mu_w = \mathbf{H}_{ww}/(2c_w)$ and $\nu_w = g_w^{(S)}/c_w$ for $w \in \{u, v\}$. Their cost-normalized conditional scores are $q_\lambda(w \mid S) = \mu_w + \lambda\nu_w$. If $\mu_u < \mu_v$ but $\nu_u > \nu_v$, their conditional scores cross once at $\lambda_{uv} = (\mu_v - \mu_u)/(\nu_u - \nu_v)$. Whenever $\lambda_{uv} \in (0,1)$, the diagonal score prefers u, whereas every $\lambda \in (\lambda_{uv}, 1]$ prefers v at the same prefix.*

Proof. The score difference is $q_\lambda(u \mid S) - q_\lambda(v \mid S) = (\mu_u - \mu_v) + \lambda(\nu_u - \nu_v)$. Under the stated inequalities it is strictly increasing from a negative value and has the unique zero λ_{uv}, which gives the two orderings. □

This is the precise sense in which a co-pruning edge changes a pruning decision: individual saliency fixes the intercept, while accumulated incident edges set the slope.

Scale invariance. For any $a > 0$, replacing $\mathbf{H}$ by $a\mathbf{H}$ multiplies every cost-normalized conditional score by the same positive constant. Hence the complete trace and stopping point are unchanged under the same guard and tie-breaking.

Proposition 7 (Finite piecewise-constant selection path)**.** *For a finite unit inventory, fixed guard and budget, and deterministic tie-breaking, the complete greedy trace is piecewise constant in $\lambda \in [0,1]$, with changes possible only at finitely many pairwise score crossings.*

Proof. At any fixed prefix, every feasible candidate score is affine in λ; two such scores can exchange order only at a crossing, unless they coincide identically, in which case deterministic tie-breaking fixes their order. There are finitely many prefixes and candidate pairs. Removing their finitely many crossing values partitions $[0, 1]$ into open intervals on which the first choice is fixed; induction fixes every subsequent prefix, the stopping point, and hence the complete trace on each interval. At the finitely many boundary points, deterministic tie-breaking defines a unique trace, possibly as a singleton piece. □

The mathematical path therefore induces a finite family of distinct removal sets that can be evaluated by measured teacher KL; Appendix A.3.2 specifies its numerical enumeration.

Budget control. Let $C = \sum_u c_u$ and $c_{\max} = \max_u c_u$. If the target is reached, the stopping rule yields $0 \le \rho_{\text{actual}} - \rho < c_{\max}/C$, equivalently $\rho C \le \sum_{u \in S} c_u < \rho C + c_{\max}$. To see this, the prefix immediately before stopping has cost below ρC; the final unit adds at most $c_{\max}$, so the stopped cost is below $\rho C + c_{\max}$.

Role of cost normalization. CoCurve greedily minimizes $\Delta\widehat{\mathcal{R}}_\lambda(u \mid S)/c_u$ over eligible candidates. Positive semidefiniteness guarantees non-negative total quadratic risk, not non-negative discrete marginals; the density rule is an efficient solver rather than a global-optimality result.

Proposition 8 (Stability of a conditional decision)**.** *Suppose* $\max_{u,v} |\mathbf{H}_{uv} - \widetilde{\mathbf{H}}_{uv}| \le \varepsilon$. *Fix the same prefix S, interaction weight $\lambda \in [0, 1]$, positive costs, and feasible pool. Writing* $q_{\mathbf{H}}(u \mid S) = (\mathbf{H}_{uu}/2 + \lambda \sum_{v \in S} \mathbf{H}_{uv})/c_u$, *we have*

$$|q_{\mathbf{H}}(u \mid S) - q_{\widetilde{\mathbf{H}}}(u \mid S)| \le \frac{\varepsilon(1/2 + \lambda|S|)}{c_u}. \tag{9}$$

If $u^\star$ uniquely minimizes $q_{\mathbf{H}}$, its choice is preserved whenever, for every other feasible v,

$$q_{\mathbf{H}}(v \mid S) - q_{\mathbf{H}}(u^\star \mid S) > \varepsilon(1/2 + \lambda|S|) \left(\frac{1}{c_v} + \frac{1}{c_{u^\star}} \right). \tag{10}$$

Proof. The first inequality follows by bounding the diagonal error and the $|S|$ accumulated edge errors separately. Applying this bound to both candidates leaves their score difference strictly positive under the stated condition. If the condition holds at every prefix of a fixed-λ run, induction gives the same complete trace and stopping point. □

The result connects estimation accuracy to the margin between actual candidates: stability depends on accumulated edge error relative to the decision gap, rather than on matrix correlation alone. The guarantee applies along each fixed-λ trace; held-out predictive risk then selects among those traces. This stability guarantee concerns perturbations of the surrogate Gram matrix, rather than agreement between surrogate decisions and the true finite-removal KL risk, which may additionally involve higher-order interactions.

B.4 What Co-Pruning Edges Add Beyond Diagonal Saliency

Remark 9 (Diagonal CoCurve recovers structured Optimal Brain Damage (OBD)). For binary removal masks, dropping off-diagonals from the implemented surrogate reduces it to $\frac{1}{2}\sum_u \mathbf{H}_{uu} s_u$, with $\mathbf{H}_{uu} = \mathbb{E}[\delta z_u^\top F^{(r)} \delta z_u] \ge 0$ and $F^{(r)} = \operatorname{diag}(p_0^{(r)}) - p_0^{(r)} p_0^{(r)\top}$ on the retained teacher support. This is a structured OBD saliency (LeCun et al., 1989) with units in place of weights and a self-distillation KL Fisher in place of the training-loss Hessian, consistent with prior Fisher, joint-removal, and interaction-aware pruning views (Theis et al., 2018; Yu et al., 2022; Kwon et al., 2022; Nonnenmacher et al., 2022). Here the first-order term vanishes exactly at $s = \mathbf{0}$ (Lemma 1), rather than through an assumption of convergence. CoCurve adds signed cross-unit curvature $\mathbf{H}_{uv}$, including attention–feed-forward network (FFN) edges, which the diagonal cannot express.

Significance. This places CoCurve on a continuum with classical second-order pruning: λ=0 is the structured KL–Fisher analogue of OBD, whereas λ=1 uses the full estimated Gram matrix. Comparing the diagonal endpoint with the selected interaction path in Table 29 therefore isolates what the co-pruning edges add while holding the unit features, guard, and budget fixed.

The exact term omitted by a diagonal method. We can quantify exactly what is lost when the same second-order surrogate $\frac{1}{2}s^\top \mathbf{H}s$ is diagonalized (Remark 4 separates that surrogate from the true risk). The statement uses the implemented Gram matrix $\mathbf{H}$; the same algebra applies to the exact curvature $\mathbf{H}^\star$.

Proposition 10 (The diagonal omission is the off-diagonal co-pruning mass)**.** *For any removal set S with indicator mask $s_S \in \{0,1\}^M$ ($s_{S,u}$=1 iff $u \in S$), the second-order co-pruning risk and its diagonal (OBD) approximation differ by exactly the off-diagonal mass of $\mathbf{H}$ on S:*

$$\Delta_{\text{off}}(S) \;:=\; \tfrac{1}{2}s_S^\top \mathbf{H}\, s_S - \tfrac{1}{2}\sum_{u\in S}\mathbf{H}_{uu} \;=\; \tfrac{1}{2}\sum_{\substack{u,v\in S\\ u\neq v}}\mathbf{H}_{uv} \;=\; \tfrac{1}{2}\sum_{\substack{u,v\in S\\ u\neq v}}\gamma_{uv}\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}, \tag{11}$$

with $\gamma_{uv} = \mathbf{H}_{uv}/\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}$ whenever the denominator is positive. By Proposition 3, γ_{uv} is exactly the cosine similarity between the stacked Fisher features $\bar{D}_u$ and $\bar{D}_v$. If either diagonal is zero, positive semidefiniteness forces $\mathbf{H}_{uv} = 0$; we set $\gamma_{uv} = 0$ for that zero-weight pair. Let $W_S = \sum_{\substack{u,v\in S\\ u\neq v}}\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}$. When $W_S > 0$, define $\bar{\gamma}_S = W_S^{-1}\sum_{\substack{u,v\in S\\ u\neq v}}\gamma_{uv}\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}$; set $\bar{\gamma}_S = 0$ when $W_S = 0$. Then $\Delta_{\text{off}}(S) = \frac{1}{2}\bar{\gamma}_S\big[(\sum_{u\in S}\sqrt{\mathbf{H}_{uu}})^2 - \sum_{u\in S}\mathbf{H}_{uu}\big]$. For the interaction-weighted surrogate $\mathbf{H}_\lambda$, the corresponding diagonal-omission term is $\lambda\Delta_{\text{off}}(S)$.

Proof. Expanding the symmetric quadratic gives $\frac{1}{2}s_S^\top\mathbf{H}s_S = \frac{1}{2}\sum_{u\in S}\mathbf{H}_{uu} + \frac{1}{2}\sum_{\substack{u,v\in S\\ u\neq v}}\mathbf{H}_{uv}$, which proves the first identity. For positive diagonal pairs, substitute $\mathbf{H}_{uv} = \gamma_{uv}\sqrt{\mathbf{H}_{uu}\mathbf{H}_{vv}}$. For a positive-semidefinite matrix, $|\mathbf{H}_{uv}|^2 \le \mathbf{H}_{uu}\mathbf{H}_{vv}$, so a zero diagonal forces the corresponding edge to vanish and the stated convention preserves the identity. Finally, when $W_S > 0$, factor out the weighted mean and use $W_S = (\sum_{u\in S}\sqrt{\mathbf{H}_{uu}})^2 - \sum_{u\in S}\mathbf{H}_{uu}$. When $W_S = 0$, every off-diagonal term vanishes by the same positive-semidefinite inequality, so both sides are zero under the stated convention. □

Significance. Within the quadratic surrogate, the omitted cross terms are themselves second order: a positive edge raises joint risk above the isolated sum, whereas a negative edge lowers it through Fisher-weighted cancellation; their aggregate magnitude is governed by $\bar{\gamma}_S$ and the removal set's saliency mass. Its discrete increment—the incident-edge sum—is precisely the additional signal that can re-rank candidates relative to the diagonal rule. The controlled comparisons in Table 29 then test whether using that signal improves the selected model while holding the unit features, guard, and budget fixed.

What the theory establishes. Teacher KL supplies a vanishing first-order term, exact gate curvature, and a cubic local remainder (Lemma 1, Proposition 2, and Proposition 5); single-unit perturbations expose a PSD graph at linear intervention cost (Proposition 3); and the graph yields an explicit ranking-reversal condition, efficient conditional marginals, a finite interaction path, budget control, and decision stability (Corollary 6, Proposition 7, and Proposition 8). The approximation diagnostics in Appendix B.2, same-budget ranking test in Appendix E.3, and support sensitivity in Appendix E.4 connect this local foundation to deployable masks. Together with Proposition 10, the chain explains both what isolated saliency omits and how CoCurve turns that missing information into conditional pruning; the matched controls in Sec. 3.6 test whether that theoretically identified signal improves finite-ratio quality under the same units, budget, and guard.

C Experimental Setup

C.1 Datasets

Table 9 lists all evaluation benchmarks with task type, metric, and split. In compact tables, A-e/A-c abbreviate ARC-easy/challenge and R-m/R-h abbreviate RACE-middle/high; HS, WG, CSQA, and OBQA denote HellaSwag, WinoGrande, CommonsenseQA, and OpenBookQA. Calibration is 128 C4 sequences (length 2048); no task labels are used for pruning. The twelve-task panel is the complete capability evaluation suite. Pruning masks are selected without downstream labels or benchmark scores.

Table 9: **Evaluation suite.** Every multiple-choice task is scored on its full split through one shared harness; n is the number of items actually scored, and for the perplexity corpora the number of target tokens under the Llama-3.1-8B-Instruct tokenizer (the token count, not the corpus, varies across tokenizers). Multiple choice is length-normalized log-likelihood over the candidate answers, zero-shot. The same taxonomy organizes the main and appendix downstream tables.

Group	**Benchmark**	**Split**	n
Language modeling	WikiText-2 (Merity et al., 2017)	test	288,936
	PTB (Marcus et al., 1993)	test	98,914
	C4 (Raffel et al., 2020)	validation, 500 docs	229,527
Commonsense	PIQA (Bisk et al., 2020)	validation	1,838
	HellaSwag (Zellers et al., 2019)	validation	10,042
	WinoGrande (Sakaguchi et al., 2021)	validation	1,267
	CommonsenseQA (Talmor et al., 2019)	validation	1,221
Science QA	ARC-easy (Clark et al., 2018)	test	2,376
	ARC-challenge (Clark et al., 2018)	test	1,172
	OpenBookQA (Mihaylov et al., 2018)	test	500
Reading	BoolQ (Clark et al., 2019)	validation	3,270
	RACE (Lai et al., 2017)	middle, test	1,436
	RACE-high (Lai et al., 2017)	high, test	3,498
	QuAIL (Rogers et al., 2020)	validation	2,164
Knowledge	MMLU (Hendrycks et al., 2021)	test	14,042

Avg_{12} is the equal-task macro average, $\bar{a} = 12^{-1} \sum_t a_t$. Its reported standard error is the task-stratified binomial estimate $100\, 12^{-1} \sqrt{\sum_t \hat{p}_t(1-\hat{p}_t)/n_t}$ in percentage points, using the item count n_t in Table 9. It quantifies finite-benchmark sampling uncertainty; the pruning rules and full-split evaluations themselves are deterministic, so it is not presented as a training-seed deviation.

C.2 Models and Experimental Protocol

All models use GQA (attention units are KV-aligned query-head groups); FFN groups are n_f=16 per layer; cost is removable parameter count. The unit definition is a property of the comparison rather than of any method: every pruning method here, ours included, chooses among the same units under the same budget.

Guard and realized budget. Every shared-protocol method protects the first and last two layers and limits each remaining layer to a removed-cost fraction $\kappa = 1.5\rho$. With equal removable cost per layer, continuous capacity is $(1 - 4/L)\min(\kappa, 1)$; for the reported 10–50% pruning-only sweep it exceeds ρ for every evaluated depth. This is a capacity check, not a guarantee of discrete feasibility. For every method using the shared budget solver, we verify the realized mask and report only runs within 0.2 percentage points of the target; baselines with a native discrete allocation retain that allocation. If the target is reached, overshoot is bounded by one unit, as established in Appendix B.3.

Guard sensitivity. At 20% on Llama-3.1-8B-Instruct, removing the guard and recalibrating selects a nearby point on its recalibrated interaction path. Without the guard, FLAP and Wanda-sp delete the first seven layers and reach WikiText perplexities of 41,189 and 30,742. The unguarded CoCurve mask shows the same early-layer concentration tendency. Thus the guard prevents extreme early-

layer concentration under a common method-independent rule. All main comparisons use the same fixed rule.

Table 10 gives the structure of the models in the pruning sweep. The 10-model mechanism panel in Appendix E additionally includes Mistral-7B (Jiang et al., 2023), Qwen2.5-7B/14B (Yang et al., 2024), Yi-1.5-9B (Young et al., 2024), Falcon3-10B (Falcon-LLM Team, 2024), and Llama-2-13B (Touvron et al., 2023). The 7-VLM analysis additionally includes Qwen2-VL-2B (Wang et al., 2024), Qwen2.5-VL-3B (Bai et al., 2025b), LLaVA-NeXT-7B (Liu et al., 2024b), and LLaVA-1.5-7B (Liu et al., 2024a). The 70B checkpoint is loaded in NF4 for all pruners in that scale study; the remaining language-model experiments use bf16 unless a quantization condition is stated explicitly.

Table 10: **Unit inventory** for the models evaluated with the shared pruning harness. The guard is one rule for every row and every ratio; nothing here is configured per model.

Model	Family	Layers	Units M	Attn/FFN
Llama-3.1-8B-Instruct	Llama-3	32	768	256/512
Llama-3.2-3B	Llama-3	28	672	224/448
Falcon3-7B	Falcon3	28	560	112/448
Mistral-Nemo-12B	Mistral	40	960	320/640
Mistral-Small-24B	Mistral	40	960	320/640
Llama-3.1-70B-Instruct	Llama-3	80	1920	640/1280

C.3 Baseline Methods

Comparison design. The main study compares selection signals under a common unit inventory, budget, guard, and evaluation harness, with weights fixed. Table 11 records each adaptation, while the common-recipe experiment separately tests the structures after lightweight recovery.

Table 11: **Pruning-only baseline configurations.** Mechanism and matched 20% outcome on Llama-3.1-8B-Instruct.

Method	Signal	Candidate damage	S-dep.	ρ	A/F	Wiki↓	Avg$_{12}$↑
Random	None	uniform	✗	20.06	34/111	21.2	46.2
Magnitude	Weights	slice norm	✗	20.03	2/126	20.5	48.1
Wanda-sp	Activations	input-weighted norm	✗	20.03	65/96	20.0	51.7
FLAP	Activations	fluctuation-weighted norm	✗	20.08	72/93	20.9	51.1
LLM-Pruner	Gradients	Taylor group score	✗	20.06	34/111	13.2	54.5
SlimGPT	Curvature	OBS reconstruction error	✗	20.07	74/92	19.2	49.7
SlimLLM	Activations	output similarity	✗	20.15	52/103	17.4	52.1
AMP	Activations	L_1 act. $\times$ weight	✗	20.00	71/93	20.4	51.7
2SSP‡	Acts. + PPL	two-stage width/depth	stage 2	20.19	0/128	16.3	54.5
CoCurve	Output deltas	node + incident edges	✓	**20.02**	**46/105**	**12.9**	**56.0**

Note. S-dep. means that a candidate's marginal cost changes with the removed set. ρ is realized parameter pruning (%); A/F counts removed attention/FFN units. All rows use the shared evaluation harness without post-pruning updates; "stage 2" denotes 2SSP's sequential attention-depth search. ‡2SSP retains its native allocation, while the others use the shared guard.

This taxonomy follows prior pruning comparisons in separating the signal, score construction, and weight-update policy (Sun et al., 2024). The final columns connect mechanism to the mask and outcome actually evaluated: all methods reach the matched parameter budget, yet they make materially different module allocations. Most baselines assign independent scores; 2SSP conditionally evaluates attention blocks only in its native depth stage, whereas CoCurve updates every candidate in one joint attention–FFN pool from its edges to all units already pruned. The asymptotic construction and solver costs are given in Appendix A.3.1; measured selection cost and its reuse across ratios are reported in Appendix D.3. We do not turn native wall-clock numbers into a ranking because published pipelines include different combinations of scoring, reconstruction, and recovery.

Wanda-sp is the structured generalization of Wanda (Sun et al., 2024) introduced by An et al. (2024). FLAP's published pipeline includes bias compensation; SlimGPT includes OBS reconstruction; SlimLLM includes output regression (An et al., 2024; Ling et al., 2024; Guo et al., 2025). The reported rows therefore compare the masks induced by their selection signals before method-specific weight updates.

Coverage and realized budgets. The primary 20–40% grid contains all seven baselines for every model from 3B through 24B. The reported 10% and 50% boundary cells are measured rather than

imputed, and every available cell passes the same realized-budget check. The 24B main blocks include all seven methods, using streamed scoring for LLM-Pruner and SlimGPT. At 70B, packed NF4 supports Random, Magnitude, Wanda-sp, FLAP, LLM-Pruner, and CoCurve; SlimGPT and SlimLLM require dense second-order state beyond the single-A100 memory contract and are the only unavailable cells at that scale.

Additional methods and controls. On Llama-3.1-8B-Instruct, Table 15 evaluates the two recent structured pruners configured in Table 11 across the full ratio sweep. AMP uses an L_1 activation–weight score (Mugnaini et al., 2025); 2SSP combines FFN-width and attention-depth pruning (Sandri et al., 2025). AMP uses the shared guard; 2SSP retains its own two-stage allocation. Single-module, diagonal, and interaction-granularity controls are reported in Appendix E.2.

C.4 Evaluation Protocol

Multiple-choice tasks are scored by length-normalized log-likelihood over the candidate answers (zero-shot, `acc_norm`); BoolQ uses the full validation split and MMLU is zero-shot log-likelihood. Perplexity uses a 2048-token context with stride 512 over the full corpus. All scores come from one shared harness.

For language models, candidate-path calibration evaluates full-vocabulary teacher KL on the 16 disjoint sequences; the top-256 support is used to construct the Fisher–Gram features, not to score candidate masks. The support-conditioned VLM path is specified below in Appendix C.5.

A sliding window gives every predicted token more left context than a non-overlapping pass, so we use the same context and stride for all pruned models and their dense reference.

C.5 Vision–Language Protocol

Units and budgets. The removable inventory contains attention heads—grouped by shared key/value head under GQA—and sixteen FFN channel groups per layer in both towers. The common budget spans this joint inventory, while the per-layer cap $\kappa = 1.5\rho$ and first/last-two-layer protection apply within each tower. We introduce no tower quota, tower weight, or additional selection hyperparameter.

Calibration. For each model, CoCurve uses 128 Flickr30k image–caption pairs (Young et al., 2014); Wanda-sp and FLAP use 64, and LLM-Pruner uses 32 for its gradient pass. Another 48 disjoint pairs select λ. These fixed counts are used for every model and drawn deterministically from one seeded calibration split. Flickr30k avoids overlap with evaluation suites built from COCO images. Only caption tokens are scored, and all unit ablations share the teacher's top-64 support. The held-out path evaluates teacher KL on the same support; no benchmark labels enter estimation or selection.

Evaluation and baselines. Each benchmark contributes 1000 items under one answer-token scoring harness, with dense models reevaluated under that harness. The structured baselines share the same inventory, budget, guard, and evaluation code. Magnitude uses all weights owned by a unit; Wanda-sp and FLAP score inputs to residual-writing projections; LLM-Pruner accumulates first-order Taylor scores one image–caption pair per backward pass; and the layer-drop control uses ShortGPT's block-influence signal.

Recovery. The language tower receives rank-128 LoRA (Hu et al., 2022) with $\alpha = 256$ while the vision encoder remains frozen and the multimodal projector is trained full-rank. At global batch 16, language and projector learning rates are 2.5×10^{-5} and 2×10^{-5}. Training uses 3000 image–instruction samples, split equally between LLaVA conversations with COCO-train/Visual Genome images (Liu et al., 2023) and short-answer TextVQA-train examples (Singh et al., 2019); training and evaluation images are disjoint.

C.6 Recovery, Quantization, and Deployment Configuration

Recovery and quantization use the complete twelve-task suite of Appendix C.1, with the same four capability groups and Avg_{12} aggregation as the pruning-only study. Their quality measurements are therefore directly comparable across deployment stages.

Table 12: **LLM recovery and numerical configuration.** These settings describe the evaluated runs; the VLM configuration is specified in Appendix C.5.

Stage	Configuration
	Estimation and selection
LLM model forwards	bf16, evaluation mode for estimation and selection
Feature store / Gram	float32 memory-mapped features and block products; float64 assembly buffer, saved as float32
Candidate solver	float64 scores and interaction sums; deterministic index ties
	Common-LoRA recovery
LoRA data	Cleaned Alpaca; one tokenizer-packed pass (589–629 optimizer steps, model dependent)
LoRA adapters	Rank 16, $\alpha = 32$, dropout 0; Q/K/V/O and gate/up/down projections; no bias training
Training	Microbatch 2, accumulation 8; AdamW, $\beta = (0.9, 0.95)$, weight decay 0, gradient clip 1
70B recovery	The same data, rank, effective batch, and schedule with microbatch 1 and accumulation 16; NF4 base weights, paged AdamW, and the fixed runtime CoCurve mask
Schedule	OneCycleLR with cosine decay, peak learning rate 10^{-4}, 3% warm-up phase; seed 0
	Weight quantization
NF4 quality test	bitsandbytes NF4 (Dettmers et al., 2023), double quantization, bf16 compute; quantize at loading, then install the fixed runtime mask
INT8 quality test	bitsandbytes LLM.int8() (Dettmers et al., 2022) with default outlier handling; install the same fixed runtime mask after loading
	Physical deployment
Inference timing	Physically sliced bf16 weights on one isolated A100-80GB with eager attention; 32 fixed prefill prompts, 16 fixed decode prompts, 128 new tokens, and 7 synchronized repeats; the same prompts are used per dense/pruned pair
Quantized timing	The same physical $30/40\%$ structures as the quality rows; bitsandbytes NF4/INT8 on the same isolated A100; prefill batch 16, decode batch 8, 128 new tokens, and 5 synchronized repeats; throughput and latency are measured in the same run

Peak memory always denotes maximum allocated CUDA memory under a fixed workload: synchronized generation for the sliced-bf16 efficiency sweep, and the forward workload for the quantized, deployment-context, and VLM comparisons. All memory and timing comparisons are made only within their shared harness.

D Complete Compression Results

The evidence follows the deployment path used in the main paper. Appendix D.1 establishes pruning-only LLM quality and locates the frontier shift, Appendix D.2 tests recovery and quantization, Appendix D.3 measures physical cost, and Appendix D.4 transfers the formulation to VLMs. Appendix D.5 changes the prompting protocol without changing the pruner, and Appendix D.6 compares pruning inherited capacity with starting from a native smaller checkpoint.

D.1 Language Modeling and Downstream Accuracy

All results use the task taxonomy in Appendix C.1. Table 13 resolves the perplexity curves numerically, and Table 14 gives the complete downstream record.

D.1.1 Perplexity Across Pruning Ratios

The pruning-only sweep covers 10% through 50%. Table 13 is the numerical counterpart of Figure 3: it reports every available method on WikiText-2, PTB, and C4 for six model slots under the common guard. The 70B block covers 10–50% under the packed-NF4 protocol. Only SlimGPT and SlimLLM are unavailable because their dense second-order state cannot be constructed on the single A100; all other 70B arms are complete.

Collapse marking. For readability, pale orange hatching marks perplexities in the practical failure regime: the model's outputs have become sufficiently degraded that fine rank distinctions no longer support a meaningful deployment comparison. The marker preserves the measured value and its within-column rank; it does not discard any result.

Table 13: **Perplexity ↓ across pruning ratios.** Six-model WikiText-2, PTB, and C4 breakdown; bold/underline mark first/second.

		Llama-3.2-3B			Falcon3-7B			Llama-3.1-8B-It			Mistral-Nemo-12B			Mistral-Small-24B			Llama-3.1-70B-It		
ρ	Method	Wiki	PTB	C4	Wiki	PTB	C4	Wiki	PTB	C4	Wiki	PTB	C4	Wiki	PTB	C4	Wiki	PTB	C4
0%	**Dense**	6.9	10.5	10.4	5.4	9.0	12.2	6.4	10.4	10.7	5.0	35.2	8.9	4.7	24.6	8.5	3.7	8.0	8.5
10%	Random	11.7	17.5	18.2	6.4	10.7	14.4	11.0	17.5	18.6	7.0	54.5	11.9	6.0	**37.0**	10.8	6.0	9.9	11.5
	Magnitude	12.3	19.8	18.6	375	462	536	9.9	15.5	17.7	13.5	207	16.9	24.9	210	39.0	5.3	9.0	10.5
	Wanda-sp	11.3	18.8	17.4	9.0	15.7	17.1	11.6	19.4	17.9	9.4	141	13.6	13.4	57.0	21.3	5.4	9.1	10.8
	FLAP	11.0	18.3	16.8	9.9	14.8	17.1	11.8	19.8	18.4	9.1	109	13.6	17.2	83.2	20.9	5.7	9.3	11.4
	LLM-Pruner	11.1	19.0	15.3	6.3	10.3	14.0	**8.8**	14.3	14.8	6.5	**51.5**	11.2	7.2	54.0	12.9	61.4	70.6	88.6
	SlimGPT	12.6	23.9	18.4	**6.0**	**10.1**	**13.5**	12.0	24.5	19.6	9.6	122	14.5	10.4	50.0	15.3	–	–	–
	SlimLLM	9.6	17.9	15.3	6.2	10.3	13.7	8.8	**13.7**	15.4	6.5	68.7	11.3	10.8	57.1	18.7	–	–	–
	CoCurve	**9.3**	**14.3**	**14.5**	6.3	10.5	14.0	8.9	20.4	**13.9**	**6.2**	52.1	**10.8**	**5.7**	52.6	**10.2**	**5.0**	**8.7**	**10.0**
20%	Random	32.5	45.9	46.6	8.3	14.2	18.5	21.2	30.4	36.8	11.7	204	18.3	**8.7**	**69.5**	15.3	8.9	14.3	16.0
	Magnitude	37.0	56.3	50.6	378	461	554	20.5	31.4	34.9	17.0	617	22.6	33.6	189	47.2	7.6	12.2	14.9
	Wanda-sp	35.2	53.8	37.4	7.9	13.4	16.8	20.0	35.3	30.1	12.1	176	19.2	14.6	275	29.3	7.4	11.7	14.3
	FLAP	40.3	81.2	50.2	7.9	13.2	16.6	20.9	36.8	29.1	15.1	478	22.3	17.0	89.6	30.8	7.9	12.4	15.3
	LLM-Pruner	20.8	31.3	27.1	7.7	12.9	16.9	13.2	**19.4**	22.0	9.2	88.3	15.0	25.8	199	47.3	425	402	730
	SlimGPT	31.8	62.8	43.5	7.4	12.6	16.5	19.2	35.5	29.2	15.2	282	22.4	23.1	123	37.8	–	–	–
	SlimLLM	16.9	29.8	24.6	11.3	17.9	20.0	17.4	30.6	29.4	11.4	187	17.8	18.1	94.0	31.4	–	–	–
	CoCurve	**16.7**	**27.3**	**23.2**	**7.1**	**11.7**	**15.6**	**12.9**	29.6	**19.6**	**8.6**	**83.4**	**14.3**	9.0	131	**14.9**	**6.4**	**9.9**	**12.4**
30%	Random	148	170	190	12.0	21.8	27.2	52.9	84.3	87.9	30.2	1047	41.4	328	1066	435	18.0	29.5	27.7
	Magnitude	318	428	472	457	529	661	324	642	274	216	1441	145	478	3159	380	14.1	24.1	28.0
	Wanda-sp	103	131	110	10.1	17.7	22.6	61.2	94.9	87.2	34.8	572	47.0	945	2307	243	12.6	21.7	26.0
	FLAP	8305	11k	3835	10.1	17.6	22.3	79.4	178	102	37.3	1105	56.2	4846	6397	3159	63.3	368	71.5
	LLM-Pruner	44.6	54.6	52.1	11.0	18.3	23.6	26.6	39.7	40.3	17.8	165	24.8	151	6060	432	918	1159	1483
	SlimGPT	111	169	142	10.1	17.8	22.6	197	317	208	58.0	715	61.0	7854	3854	5390	–	–	–
	SlimLLM	42.7	79.0	**50.5**	15.3	24.2	25.6	78.2	101	109	33.3	460	39.5	78.8	465	116	–	–	–
	CoCurve	**34.8**	**49.4**	55.0	**8.9**	**15.1**	**19.5**	**21.6**	**36.3**	**32.0**	**15.8**	**160**	**23.3**	**20.5**	**419**	**29.7**	**8.4**	**13.6**	**17.2**
40%	Random	13k	9552	4957	20.4	36.9	45.6	4168	3201	3152	197	6702	262	3299	5950	2847	87.6	89.9	91.1
	Magnitude	13k	8180	8512	485	779	618	27k	24k	14k	237k	163k	192k	7766	12k	5370	915	1216	794
	Wanda-sp	701	592	655	29.9	52.4	53.1	1015	1288	861	479	8909	550	4809	4747	1271	315	768	633
	FLAP	11k	19k	6720	31.4	51.8	55.4	50k	30k	14k	834	16k	545	5340	6506	4034	1396	4245	1816
	LLM-Pruner	165	316	204	18.4	28.7	37.9	103	165	126	73.1	499	86.3	5495	8725	4551	767	1483	1320
	SlimGPT	8584	9960	6614	22.9	40.7	46.4	26k	32k	13k	84k	51k	2786	9057	11k	7197	–	–	–
	SlimLLM	**136**	**160**	**130**	58.4	102	57.5	184	211	217	144	1132	146	3969	11k	6273	–	–	–
	CoCurve	194	257	196	**13.8**	**23.1**	**29.2**	**59.1**	**108**	**76.1**	**52.2**	**455**	**71.0**	**277**	**4037**	**232**	**33.3**	**39.8**	**58.1**
50%	Random	31k	34k	23k	42.6	76.4	93.7	134k	102k	41k	2164	122k	1614	6471	13k	5636	7831	8050	5323
	Magnitude	128k	76k	177k	660	1312	905	165k	332k	185k	283k	276k	278k	9366	16k	9551	33k	37k	40k
	Wanda-sp	60k	79k	47k	818	858	588	370k	454k	34k	691k	200k	414k	6249	10k	4769	18k	27k	21k
	FLAP	121k	181k	132k	753	1317	518	627k	328k	506k	728k	322k	1203k	6525	9779	5484	19k	22k	16k
	LLM-Pruner	4003	8385	1882	57.9	117	95.9	865	2408	1182	**513**	**1621**	434	9320	12k	8449	14k	17k	7805
	SlimGPT	111k	149k	119k	1087	1428	644	109k	155k	135k	1957k	7904k	1023k	6136	8956	5027	–	–	–
	SlimLLM	**1199**	**945**	**1138**	163	481	176	**773**	**705**	**611**	673	3115	**407**	4666	**7828**	3936	–	–	–
	CoCurve	4617	6180	3355	**41.7**	**61.7**	**67.0**	956	1804	829	839	2576	643	**2325**	9330	**1874**	**713**	**1741**	**792**

Equal-perplexity compression. Interpolating the baseline envelope linearly in log-perplexity between measured ratios, without extrapolation, gives a complementary view of the curve. At 30%, CoCurve permits 2.2–6.1 percentage points more pruning at the same WikiText-2 perplexity on all five completed models. At 40%, four of five gain 2.4–6.6 points, while the 3B model trails the interpolated envelope by 1.6 points. These are interpolated comparisons within the measured sweep, not additional evaluated masks.

Mistral-Small-24B gives the largest high-ratio margin in the completed sweep. At $\rho = 30\%$, CoCurve reaches WikiText-2 perplexity 20.5 against 78.8 for the strongest baseline, a 3.8× ratio; at 40%, the ratio reaches 11.9× (Table 13).

D.1.2 Downstream Accuracy

Table 14 gives all twelve downstream tasks for every method at 20–40%, plus the measured 50% boundary on the 7B–24B models.

Table 14: **Full downstream accuracy by task.** All five language models and every method in the matched grid; the 7B–24B models additionally report the 50% boundary.

			Commonsense ↑				Science QA ↑			Reading ↑				Know. ↑	
Model	ρ	**Method**	**PIQA**	**HellaS**	**WinoG**	**CSQA**	**ARC-e**	**ARC-c**	**OBQA**	**BoolQ**	**RACE**	**RACE-h**	**QuAIL**	**MMLU**	**Avg$_{12}$**
3B	–	**Dense**	77.4	72.8	58.6	57.8	71.3	46.9	43.2	74.3	59.9	45.5	54.9	41.8	58.7±0.35
	20%	Random	63.1	42.4	51.5	32.1	45.0	27.1	28.6	44.5	37.7	29.4	39.7	28.8	39.2±0.34
		Magnitude	66.0	46.8	**54.2**	37.9	50.5	31.8	31.8	45.8	42.8	33.8	44.1	31.6	43.1±0.34
		Wanda-sp	68.9	48.6	51.6	37.7	53.5	30.1	31.6	59.3	41.4	34.0	43.0	30.7	44.2±0.34
		FLAP	67.1	46.4	52.7	36.5	51.4	27.0	28.8	63.9	40.1	31.8	42.1	30.2	43.2±0.34
		LLM-Pruner	70.6	55.3	53.4	42.3	55.2	32.7	**34.6**	60.8	43.2	35.0	42.4	33.5	46.6±0.35
		SlimGPT	69.3	45.7	52.0	33.1	49.7	27.8	30.6	63.0	37.4	30.7	39.7	28.8	42.3±0.34
		SlimLLM	70.3	52.9	51.4	40.7	54.2	32.3	32.8	46.8	43.5	34.4	44.1	32.0	44.6±0.34
		CoCurve	**70.6**	**55.7**	54.1	**45.9**	**57.7**	**35.2**	34.4	**65.6**	**49.3**	**37.8**	**46.2**	**33.9**	**48.9**±0.35
	30%	Random	58.8	31.0	49.7	24.0	34.7	23.9	24.8	39.0	29.3	27.4	36.9	27.1	33.9±0.32
		Magnitude	57.5	31.5	51.2	28.4	34.3	25.4	23.8	48.3	30.2	26.8	33.6	27.2	34.8±0.33
		Wanda-sp	59.7	33.1	49.7	29.5	42.2	23.0	26.0	51.7	32.9	27.1	35.8	27.3	36.5±0.33
		FLAP	54.2	27.3	49.3	24.7	32.1	23.6	24.8	39.7	27.8	25.8	26.5	26.8	31.9±0.32
		LLM-Pruner	64.4	**42.5**	**52.6**	34.4	44.7	27.0	29.4	**62.3**	39.3	31.6	40.7	29.5	41.5±0.34
		SlimGPT	59.2	32.7	51.0	24.7	35.7	23.6	26.2	56.3	33.7	26.9	36.2	26.8	36.1±0.33
		SlimLLM	**64.6**	40.0	50.3	33.3	45.3	26.4	27.8	38.5	35.3	29.9	38.7	28.8	38.2±0.33
		CoCurve	63.7	42.3	51.2	**35.6**	**45.8**	**27.7**	**32.2**	61.1	**39.7**	**31.9**	**44.0**	**30.0**	**42.1**±0.34
	40%	Random	52.7	26.7	50.5	23.4	28.8	24.4	24.2	40.1	26.7	24.7	30.1	26.5	31.6±0.32
		Magnitude	51.3	24.9	48.2	21.0	27.6	25.6	23.2	38.1	26.2	25.9	24.2	27.0	30.3±0.32
		Wanda-sp	54.2	27.0	48.4	24.2	32.3	20.2	22.8	45.4	28.4	26.4	27.7	26.2	31.9±0.32
		FLAP	50.9	25.3	49.4	21.4	27.3	**25.8**	23.2	41.4	25.3	25.2	25.3	26.3	30.6±0.32
		LLM-Pruner	59.0	**35.0**	**51.7**	29.0	37.8	25.6	**27.6**	**60.6**	32.8	**28.8**	**36.6**	28.0	**37.7**±0.33
		SlimGPT	50.2	25.6	48.2	22.5	29.8	22.5	25.6	38.8	26.5	25.6	25.5	26.2	30.6±0.32
		SlimLLM	**61.0**	30.9	49.5	**29.4**	**39.9**	22.5	24.8	41.0	32.5	26.6	35.6	**28.0**	35.2±0.33
		CoCurve	55.9	30.1	49.6	25.2	34.3	24.4	26.8	53.3	**33.6**	28.2	36.4	26.8	35.4±0.33
F3-7B	–	**Dense**	79.5	71.9	64.6	49.7	77.4	53.9	39.8	82.1	64.5	47.4	52.7	45.2	60.7±0.34
	20%	Random	74.6	61.5	58.1	43.3	**67.6**	41.6	32.0	**73.8**	52.6	40.7	48.3	38.1	52.7±0.34
		Magnitude	53.5	27.1	51.0	21.8	31.1	22.1	23.8	39.9	28.8	26.0	28.7	26.7	31.7±0.32
		Wanda-sp	74.4	63.8	58.0	**43.9**	65.2	42.4	32.4	72.0	55.8	42.8	49.4	38.9	53.3±0.34
		FLAP	74.7	**64.2**	56.8	43.6	67.5	42.3	**35.2**	72.8	**56.1**	41.7	48.4	38.2	**53.5**±0.34
		LLM-Pruner	**74.8**	62.9	56.9	43.5	67.6	**42.7**	31.8	69.9	55.1	**43.0**	**50.1**	**39.8**	53.2±0.34
		SlimGPT	74.0	62.9	**59.3**	40.7	66.2	38.3	33.6	73.1	55.4	41.9	48.8	38.8	52.8±0.34
		SlimLLM	74.3	62.4	54.2	42.9	63.9	41.2	33.4	65.7	45.1	37.8	42.3	36.8	50.0±0.34
		CoCurve	73.0	63.2	57.9	43.7	**67.6**	41.7	34.8	70.8	53.3	41.2	49.8	37.8	52.9±0.35
	30%	Random	68.4	55.3	**57.1**	38.9	57.2	**38.1**	26.4	**65.8**	46.3	36.9	44.8	33.6	47.4±0.34
		Magnitude	54.7	28.3	49.4	23.6	34.0	22.6	23.8	42.4	28.8	26.9	27.8	27.4	32.5±0.32
		Wanda-sp	66.7	52.3	54.9	33.6	53.3	32.0	28.8	65.1	46.0	36.8	42.8	33.2	45.5±0.34
		FLAP	69.0	52.4	54.5	38.2	58.8	34.5	29.4	62.9	45.8	37.2	43.2	33.5	46.6±0.34
		LLM-Pruner	**70.5**	55.4	56.6	38.7	59.4	34.0	29.6	65.6	**51.0**	38.2	**47.6**	34.8	48.5±0.34
		SlimGPT	68.6	53.1	53.7	38.1	55.3	33.9	28.4	64.6	47.0	36.0	43.5	33.3	46.3±0.34
		SlimLLM	**70.5**	**57.2**	52.6	37.7	56.2	36.4	29.2	65.6	44.3	34.6	37.9	34.6	46.4±0.34
		CoCurve	69.3	57.1	56.1	**42.5**	**61.3**	35.2	**31.2**	64.0	48.5	**38.5**	46.0	**35.8**	**48.8**±0.34
	40%	Random	62.7	43.4	51.9	31.4	45.9	29.7	23.6	49.0	39.6	33.3	42.8	29.8	40.3±0.33
		Magnitude	52.9	27.1	49.8	24.4	32.0	22.4	23.6	47.6	29.7	26.7	23.4	27.3	32.2±0.32
		Wanda-sp	59.9	39.0	50.0	28.6	43.4	27.0	26.2	47.3	33.4	30.1	35.4	29.1	37.5±0.33
		FLAP	57.8	39.3	50.5	28.3	41.3	24.8	23.0	48.3	34.5	29.5	33.3	28.7	36.6±0.33
		LLM-Pruner	**65.2**	46.1	52.6	**34.9**	49.7	28.8	27.0	62.9	**44.1**	**34.7**	**44.4**	31.1	**43.5**±0.34
		SlimGPT	61.7	40.4	51.5	30.5	45.0	25.7	25.6	50.3	36.8	31.3	38.8	29.7	38.9±0.33
		SlimLLM	63.5	43.5	50.6	27.8	47.5	29.3	**27.6**	59.3	34.5	29.8	30.5	30.0	39.5±0.33
		CoCurve	63.8	**46.3**	**53.1**	34.7	**50.1**	**31.0**	27.2	**63.2**	42.1	33.9	42.1	**31.6**	43.3±0.34
	50%	Random	57.1	34.7	50.8	25.7	37.2	24.6	24.8	48.1	35.7	**30.8**	**38.4**	27.4	36.3±0.33
		Magnitude	52.3	26.7	50.9	23.8	31.7	23.4	24.8	40.9	28.8	25.3	27.8	27.5	32.0±0.32
		Wanda-sp	52.4	27.7	50.6	22.8	29.6	22.3	**27.8**	41.8	29.8	27.1	29.3	27.0	32.3±0.32
		FLAP	52.4	28.1	50.9	24.5	29.6	22.2	24.2	43.4	27.6	28.0	27.2	26.8	32.1±0.32
		LLM-Pruner	**60.2**	35.5	49.4	**29.4**	**37.5**	**25.3**	24.2	47.3	**36.2**	29.6	34.9	**28.9**	36.5±0.33
		SlimGPT	52.8	27.5	50.4	21.9	28.9	23.8	25.6	43.2	27.6	27.1	29.3	27.0	32.1±0.32
		SlimLLM	60.0	32.1	50.9	23.3	35.9	25.1	24.6	**59.1**	28.6	27.0	22.9	27.5	34.7±0.32
		CoCurve	59.4	**35.7**	**51.9**	25.9	36.9	24.9	27.6	57.2	35.0	30.4	38.1	28.5	**37.6**±0.33
8B-It	–	**Dense**	80.9	77.4	66.2	59.2	78.1	54.4	47.6	85.0	68.8	54.1	62.7	47.8	65.2±0.34
	20%	Random	68.2	53.8	54.5	38.5	50.3	29.4	31.4	65.4	47.1	36.1	47.8	31.8	46.2±0.34
		Magnitude	68.9	55.5	55.9	41.7	54.6	31.4	32.4	65.2	50.2	39.1	49.4	32.8	48.1±0.34
		Wanda-sp	73.7	57.4	57.0	48.9	62.5	38.8	36.4	72.2	52.8	38.7	46.6	35.6	51.7±0.35
		FLAP	73.2	59.0	54.9	46.7	62.5	37.7	38.2	70.9	49.9	39.2	45.6	35.3	51.1±0.35
		LLM-Pruner	75.0	64.6	57.4	**51.8**	60.4	37.2	36.8	**79.1**	57.2	**45.2**	52.4	36.8	54.5±0.34
		SlimGPT	73.4	58.8	56.6	44.6	61.9	37.4	35.4	64.3	47.8	36.6	44.5	35.1	49.7±0.35
		SlimLLM	75.2	60.0	56.2	47.6	**65.2**	39.5	**39.8**	65.8	51.3	39.4	47.8	37.2	52.1±0.35

continued on the next page

Model	ρ	Method	Commonsense ↑				Science QA ↑			Reading ↑				Know. ↑	Avg_{12}
			PIQA	HellaS	WinoG	CSQA	ARC-e	ARC-c	OBQA	BoolQ	RACE	RACE-h	QuAIL	MMLU	
		CoCurve	**75.8**	**67.5**	**61.0**	50.4	62.5	**41.9**	39.6	78.3	**58.1**	44.0	**54.1**	**38.5**	**56.0**±0.35
	30%	Random	60.6	40.2	53.5	29.6	41.0	24.1	26.6	47.8	41.4	31.8	42.6	28.6	39.0±0.33
		Magnitude	57.0	32.2	51.0	24.7	31.5	25.0	25.6	51.3	33.1	28.8	34.8	27.3	35.2±0.33
		Wanda-sp	65.7	38.5	53.4	33.7	45.8	27.8	28.2	47.7	41.9	31.0	40.1	29.6	40.3±0.34
		FLAP	65.7	41.0	53.6	33.5	43.8	24.8	26.8	51.2	37.8	32.0	37.2	29.0	39.7±0.33
		LLM-Pruner	**71.0**	54.4	52.4	**43.2**	51.0	33.2	30.4	**73.4**	**48.4**	**38.8**	47.8	31.9	48.0±0.34
		SlimGPT	62.1	38.9	51.2	30.4	42.8	25.4	27.4	51.3	35.4	29.4	38.0	27.9	38.4±0.33
		SlimLLM	69.5	37.8	52.0	36.9	53.0	28.8	28.2	62.5	35.9	29.4	37.2	30.4	41.8±0.34
		CoCurve	70.5	**55.8**	**55.5**	41.0	**55.6**	**34.4**	**35.2**	73.1	46.5	37.3	**50.1**	**33.4**	**49.0**±0.35
	40%	Random	54.0	29.0	50.5	24.1	31.3	23.4	24.0	42.5	33.3	26.4	35.4	26.7	33.4±0.32
		Magnitude	51.1	25.4	50.8	19.0	25.7	25.9	27.8	48.2	25.6	25.8	24.4	26.2	31.3±0.32
		Wanda-sp	54.8	26.3	48.2	25.6	33.1	22.4	22.8	38.7	27.8	26.0	24.8	26.6	31.4±0.32
		FLAP	52.1	26.0	49.2	22.0	28.0	23.5	22.6	41.6	26.6	25.4	23.3	25.6	30.5±0.32
		LLM-Pruner	**65.3**	**42.7**	**52.2**	**33.4**	40.0	28.1	26.8	**65.1**	36.0	32.2	40.2	**29.7**	41.0±0.33
		SlimGPT	51.9	26.2	49.2	21.0	27.9	23.9	29.2	38.2	25.3	26.7	21.1	26.9	30.6±0.32
		SlimLLM	62.6	31.2	50.4	29.2	**41.6**	24.0	25.0	51.2	31.6	27.3	32.1	27.6	36.1±0.33
		CoCurve	64.0	41.0	51.4	31.9	41.6	**28.7**	**29.6**	62.3	**41.9**	**32.2**	**42.8**	29.3	**41.4**±0.34
	50%	Random	50.8	24.3	48.0	21.2	26.9	24.1	22.8	39.5	26.0	26.5	26.6	26.6	30.3±0.32
		Magnitude	50.0	25.7	48.3	18.6	27.0	**27.5**	25.2	58.3	24.5	24.2	22.5	24.7	31.4±0.32
		Wanda-sp	51.8	26.1	48.6	20.0	24.7	21.5	21.2	43.2	25.1	24.7	**28.5**	25.5	30.1±0.31
		FLAP	49.8	24.8	49.3	20.4	24.4	26.3	21.2	55.9	25.1	25.1	27.2	25.8	31.3±0.31
		LLM-Pruner	**58.0**	**33.2**	**51.1**	**25.7**	32.7	26.1	24.2	41.4	29.7	**29.4**	24.4	**27.9**	33.7±0.32
		SlimGPT	50.7	25.8	50.0	20.1	26.2	24.1	**26.6**	58.4	23.8	24.1	27.9	25.3	31.9±0.32
		SlimLLM	56.0	26.9	49.6	23.5	**33.6**	20.5	24.0	52.2	28.6	25.7	17.4	26.5	32.0±0.32
		CoCurve	55.7	30.0	50.4	24.2	30.7	26.5	24.8	**60.8**	**31.1**	27.8	24.5	27.5	**34.5**±0.32
Nemo-12B	–	**Dense**	82.3	80.0	72.1	53.8	81.0	59.5	48.2	85.1	66.3	50.2	57.4	47.4	65.3±0.34
	20%	Random	75.6	64.9	60.4	39.6	59.4	37.8	35.8	63.1	49.3	38.1	50.4	35.6	50.8±0.34
		Magnitude	74.1	61.4	60.5	42.2	59.0	36.5	33.0	64.3	49.1	37.4	48.4	34.6	50.1±0.34
		Wanda-sp	74.8	63.8	56.5	41.2	62.8	41.8	35.4	65.7	48.4	38.0	47.3	35.5	50.9±0.35
		FLAP	72.7	63.5	55.0	39.1	59.2	36.6	33.0	63.1	47.4	36.2	43.9	35.4	48.8±0.34
		LLM-Pruner	76.2	**67.4**	**62.9**	**46.4**	**70.7**	**46.0**	37.2	64.1	**57.1**	**42.5**	51.6	**37.6**	55.0±0.35
		SlimGPT	74.2	62.4	59.1	36.9	54.8	35.6	33.6	64.1	43.3	35.2	45.8	33.7	48.2±0.34
		SlimLLM	74.4	62.3	57.9	41.3	63.8	39.5	35.4	68.2	50.8	39.1	47.7	35.0	51.3±0.35
		CoCurve	**76.3**	66.5	**62.9**	42.7	66.5	43.3	**39.4**	**79.1**	54.1	41.0	**52.6**	36.9	**55.1**±0.34
	30%	Random	66.1	51.8	54.1	33.0	47.2	29.5	30.2	48.5	41.0	32.4	44.9	30.2	42.4±0.34
		Magnitude	58.1	34.9	50.4	24.4	34.0	24.3	27.2	58.8	32.2	28.8	33.6	27.3	36.2±0.33
		Wanda-sp	66.2	46.5	51.5	31.0	48.6	28.6	29.4	62.1	42.1	32.8	42.1	30.2	42.6±0.34
		FLAP	64.0	45.6	52.0	29.3	42.0	27.2	29.6	56.5	38.5	31.8	39.5	29.4	40.5±0.34
		LLM-Pruner	**72.0**	**57.6**	**58.7**	**40.6**	**59.9**	**37.3**	**31.4**	**64.9**	**50.7**	**38.5**	48.0	**33.0**	**49.4**±0.34
		SlimGPT	64.4	44.5	53.2	30.9	41.6	28.7	**31.4**	56.9	34.1	30.6	39.3	29.0	40.4±0.34
		SlimLLM	68.2	46.3	54.1	34.2	53.6	31.3	28.8	62.1	40.3	32.0	40.1	30.8	43.5±0.34
		CoCurve	71.4	56.7	57.5	38.9	56.7	35.3	30.4	62.9	48.7	37.3	**48.7**	31.7	48.0±0.34
	40%	Random	57.3	34.1	49.6	27.0	35.4	22.8	24.6	58.0	33.1	27.9	33.6	27.0	35.9±0.33
		Magnitude	49.4	26.0	48.2	21.0	25.0	25.4	25.6	38.4	24.5	26.3	23.8	26.7	30.0±0.32
		Wanda-sp	56.3	29.3	49.9	26.0	35.7	23.3	22.8	59.1	28.8	27.9	28.8	27.2	34.6±0.32
		FLAP	53.8	29.8	48.9	24.4	29.1	23.4	24.6	55.1	28.2	27.3	27.8	26.3	33.2±0.32
		LLM-Pruner	**63.4**	**43.1**	**54.5**	**34.3**	**47.9**	28.3	**28.8**	**62.2**	**43.7**	**34.7**	**43.1**	**30.0**	**42.8**±0.34
		SlimGPT	52.8	28.1	48.2	20.1	28.9	22.8	24.8	52.3	29.0	26.4	28.4	26.2	32.3±0.32
		SlimLLM	58.7	30.8	50.0	26.7	38.7	24.1	24.0	53.4	31.7	26.8	32.6	27.5	35.4±0.33
		CoCurve	62.7	42.1	53.7	34.2	43.8	**29.4**	24.2	62.1	41.9	32.5	41.6	29.3	41.5±0.33
	50%	Random	52.5	27.5	50.4	22.8	29.6	23.3	26.0	50.9	29.0	25.6	26.7	26.1	32.5±0.32
		Magnitude	50.6	25.3	48.7	20.1	25.1	25.4	26.2	55.7	24.4	25.0	25.3	24.0	31.3±0.32
		Wanda-sp	50.8	25.2	48.7	19.5	26.1	25.0	20.4	44.7	24.7	24.7	25.9	25.9	30.1±0.31
		FLAP	52.0	25.7	49.3	20.6	25.6	25.3	24.6	43.2	26.9	23.0	24.4	25.1	30.5±0.32
		LLM-Pruner	**57.1**	**34.2**	50.4	**30.2**	**39.6**	**27.2**	25.2	54.9	**34.4**	**31.5**	**30.5**	**28.0**	**36.9**±0.33
		SlimGPT	50.9	25.5	**50.7**	19.7	26.2	24.7	**27.6**	47.3	25.3	24.3	21.7	26.2	30.8±0.32
		SlimLLM	55.5	26.5	48.9	23.8	31.0	23.7	22.2	38.1	28.0	25.6	26.5	26.7	31.4±0.32
		CoCurve	53.4	30.8	49.6	24.2	34.1	24.2	21.6	**60.3**	32.5	28.1	25.6	27.6	34.3±0.32
24B	–	**Dense**	83.0	81.6	75.1	54.5	82.6	59.6	48.0	86.8	68.4	51.0	57.3	52.3	66.7±0.33
	20%	Random	76.9	69.3	61.9	50.5	72.9	47.0	39.0	75.7	60.2	43.6	52.6	43.2	57.7±0.34
		Magnitude	76.7	67.5	60.7	49.1	72.9	45.7	40.6	73.2	58.6	42.4	52.2	42.4	56.8±0.35
		Wanda-sp	73.2	70.8	59.9	46.8	66.7	45.5	41.4	**81.2**	58.1	41.2	53.7	41.0	56.6±0.35
		FLAP	71.6	68.9	59.6	46.4	65.2	44.8	40.4	78.8	56.5	41.5	51.5	40.6	55.5±0.35
		LLM-Pruner	74.2	73.5	65.2	49.1	73.1	51.1	41.4	78.5	58.8	**47.0**	55.1	**43.7**	59.2±0.34
		SlimGPT	76.2	71.5	61.5	47.9	69.9	46.1	40.0	74.7	60.5	42.5	52.1	41.4	57.0±0.35
		SlimLLM	**78.0**	74.2	65.0	50.7	73.8	49.5	40.4	78.6	58.9	44.9	52.9	43.5	59.2±0.34
		CoCurve	77.8	**76.0**	**66.3**	**51.2**	**74.1**	**52.7**	**42.0**	79.7	**63.6**	44.3	**55.9**	43.5	**60.6**±0.34
	30%	Random	70.2	54.5	57.7	44.4	59.4	37.1	33.8	42.2	53.8	38.0	47.3	35.5	47.8±0.35
		Magnitude	62.6	50.3	54.1	33.5	52.6	34.9	29.4	66.1	42.9	34.7	43.6	31.7	44.7±0.34
		Wanda-sp	63.2	50.5	51.5	37.3	55.6	36.0	33.6	59.4	41.8	33.3	37.2	33.7	44.4±0.35
		FLAP	52.3	26.8	49.4	20.0	25.8	24.2	23.0	37.8	27.3	26.8	20.0	26.5	30.0±0.32
		LLM-Pruner	62.8	59.4	58.0	38.2	55.7	38.2	31.4	**70.8**	50.2	38.6	46.3	33.7	48.6±0.34
		SlimGPT	63.1	48.9	52.5	33.0	48.9	30.7	32.4	59.6	39.2	30.6	40.5	31.4	42.6±0.34
		SlimLLM	70.8	61.5	54.7	**47.1**	**67.0**	**42.3**	**40.0**	63.4	51.3	39.1	47.0	36.5	51.7±0.35
		CoCurve	**72.0**	**63.8**	**58.8**	42.4	63.8	42.2	38.4	69.9	**55.2**	**40.0**	**51.4**	**36.5**	**52.9**±0.35
	40%	Random	57.2	36.3	51.6	25.6	34.9	28.4	26.2	38.2	31.2	28.3	31.6	27.9	34.8±0.33
		Magnitude	55.0	27.9	50.4	24.3	32.6	27.6	29.4	50.6	30.7	27.4	25.0	27.2	34.0±0.33
		Wanda-sp	52.4	28.1	49.6	22.0	29.7	22.0	25.6	38.6	27.3	27.2	25.8	26.8	31.3±0.32
		FLAP	52.2	27.7	49.4	22.7	29.3	23.5	26.2	39.5	28.4	26.4	23.2	26.8	31.3±0.32
		LLM-Pruner	55.0	34.4	52.5	23.5	35.9	27.0	26.4	38.3	30.0	28.6	27.9	28.5	34.0±0.33
		SlimGPT	51.3	25.3	50.4	21.9	25.4	23.0	26.8	37.8	26.0	26.3	24.2	26.8	30.5±0.32
		SlimLLM	**58.9**	39.0	52.6	**30.1**	**47.7**	**32.3**	**32.8**	46.9	34.3	30.0	25.2	30.0	38.3±0.34
		CoCurve	56.0	**39.9**	**54.1**	27.2	41.0	30.1	28.2	**51.3**	**40.5**	**33.1**	**42.2**	**30.0**	**39.5**±0.34
	50%	Random	**53.4**	27.9	49.1	19.8	27.7	24.8	28.8	37.8	26.0	25.7	25.4	27.2	31.1±0.32
		Magnitude	51.8	25.5	49.3	20.5	27.7	**25.3**	26.6	38.6	25.3	24.9	23.6	26.0	30.4±0.32
		Wanda-sp	50.5	25.3	47.8	19.1	25.9	23.5	26.2	37.8	26.5	25.4	20.9	27.0	29.7±0.32
		FLAP	48.4	25.5	49.8	20.3	25.3	24.3	**29.2**	37.8	25.8	26.1	24.3	26.9	30.3±0.32
		LLM-Pruner	51.7	26.8	49.3	22.0	28.7	23.5	27.8	38.1	28.0	27.0	23.9	27.0	31.2±0.32
		SlimGPT	49.1	25.4	49.8	20.2	26.0	22.9	25.0	38.7	26.2	24.6	29.8	25.9	30.3±0.32
		SlimLLM	48.2	25.8	**50.5**	20.0	25.4	23.9	29.0	37.9	25.6	25.3	**30.2**	26.4	30.7±0.32
		CoCurve	52.6	**30.1**	48.8	**22.4**	**33.2**	24.9	24.0	**46.6**	**34.1**	**29.2**	24.2	**28.2**	**33.2**±0.32

Across the 19 complete model–ratio blocks, CoCurve ranks first on Avg_{12} in 13 and first or second in 18. The coverage is not confined to one architecture or an easy ratio: every completed block from 30% through 50% is top-two, spanning all five models represented in the table. Together with the corpus-level sweep above, this shows that the gain transfers from next-token prediction to a broad capability profile rather than being carried by one task family.

D.1.3 Additional Method and Granularity Coverage

Table 15: **Additional pruning methods across** 20–50% on Llama-3.1-8B-Instruct. AMP uses the shared guard; ‡2SSP retains its native two-stage allocation. All twelve tasks use exactly the scoring and aggregation rules of Table 1 and Appendix C.1.

		Perplexity ↓			Downstream accuracy (%) ↑											
ρ	Method	Wiki	PTB	C4	PIQA	HS	WG	CSQA	A-e	A-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
–	**Dense**	6.40	10.36	10.67	80.9	77.4	66.2	59.2	78.1	54.4	47.6	85.0	68.8	54.1	62.7	47.8	65.2
20%	AMP	20.39	47.27	33.26	74.7	58.1	56.7	49.2	63.9	38.8	38.8	69.4	51.1	39.5	44.1	35.7	51.7
	2SSP‡	16.29	**24.07**	29.00	73.6	61.9	59.7	49.4	**66.8**	41.3	38.6	64.3	**60.0**	**45.9**	**54.2**	37.9	54.5
	CoCurve	**12.88**	29.63	**19.58**	**75.8**	**67.5**	**61.0**	**50.4**	62.5	**41.9**	**39.6**	**78.3**	58.1	44.0	54.1	**38.5**	**56.0**
30%	AMP	87.89	158.9	121.3	66.8	37.5	51.9	40.7	52.2	29.1	30.0	45.6	36.9	29.7	38.4	30.1	40.7
	2SSP‡	43.63	73.44	70.37	68.0	47.0	53.8	37.5	51.2	31.5	32.2	63.7	**48.8**	35.7	44.4	31.8	45.5
	CoCurve	**21.59**	**36.31**	**31.99**	**70.5**	**55.8**	**55.5**	**41.0**	**55.6**	**34.4**	**35.2**	**73.1**	46.5	**37.3**	**50.1**	**33.4**	**49.0**
40%	AMP	1.3e3	2.0e3	1.0e3	55.2	26.5	48.5	26.5	34.3	23.7	23.4	38.0	27.6	26.0	21.7	27.3	31.6
	2SSP‡	292.2	508.2	334.3	62.1	34.7	50.9	27.6	38.2	25.8	27.8	62.0	39.7	30.4	38.2	28.0	38.8
	CoCurve	**59.13**	**108.4**	**76.05**	**64.0**	**41.0**	**51.4**	**31.9**	**41.6**	**28.7**	**29.6**	**62.3**	**41.9**	**32.2**	**42.8**	**29.3**	**41.4**
50%	AMP	3.6e4	5.4e4	2.2e4	52.8	25.1	48.6	20.4	27.3	22.5	21.8	39.9	26.7	26.2	27.0	26.4	30.4
	2SSP‡	2.6e3	2.3e3	1.9e3	53.5	26.5	49.0	22.0	29.8	22.1	**25.6**	53.1	30.8	26.6	**30.8**	26.6	33.0
	CoCurve	**956.4**	**1.8e3**	**828.8**	**55.7**	**30.0**	**50.4**	**24.2**	**30.7**	**26.5**	24.8	**60.8**	**31.1**	**27.8**	24.5	**27.5**	**34.5**

The comparison preserves AMP's shared-budget protocol and 2SSP's native allocation. Across 20–50%, CoCurve leads Avg_{12} at all four ratios and 11 of the 12 corpus-level perplexity columns; the sole exception is PTB at 20%, where it ranks second. The result extends the advantage to a wider method pool under the same evaluation contract.

Finer-grained baseline units. Wanda-sp, FLAP, and Magnitude are also evaluated with 64 FFN groups per layer on Llama-3.1-8B-Instruct at 20%, preserving the guard and evaluation harness. Wanda-sp improves from 20.02 to 14.86 WikiText-2 perplexity and FLAP from 20.88 to 16.23, while Magnitude declines on four of the five summary metrics. CoCurve with 16 groups retains lower perplexity and higher accuracy in every capability group than these finer-grained arms. This control shows that its advantage is not explained by giving the baselines a coarser FFN inventory.

D.1.4 Where CoCurve Advances the Deployment Frontier

Frontier criterion. The quality–deployment frontier is an operating curve, not a single pruning ratio. A method advances it by retaining more quality at a fixed resource budget or by reaching the same quality with fewer resources. We test both directions and then connect them to measured memory and throughput (Figure 10).

Quality at a fixed pruning budget. Across the 6 LLMs and 3 corpora, CoCurve wins 83%, 94%, and 83% of the 18 matched comparisons at 20/30/40% pruning. More importantly, the median strongest-baseline-to-CoCurve perplexity ratio grows from 1.06× to 1.16× and 1.36× as compression tightens. The benefit is smallest while every pruner retains ample redundancy and becomes most visible as baseline curves begin to bend.

Resources at matched quality. Matched-quality interpolation gives the deployment meaning of that separation. On the five-model shared grid, CoCurve supports 2.2–6.6 additional pruning points in 9/10 model–ratio comparisons at 30/40%; the sole exception is the 3B model at 40%, where its small capacity has already entered a brittle regime. The largest shifts occur on the 24B model, but

positive results on Falcon, Llama, and Mistral-Nemo show that scale alone is not the condition. The gain is strongest near the curve's bend, where usable redundancy remains but isolated scores begin to miss compatible removals.

Measured physical path. At a fixed pruning ratio, all sliced selectors occupy a narrow throughput–memory band (Table 18), so CoCurve's gain comes from retaining more quality at essentially the same physical footprint; moving to a higher ratio then converts that quality margin into additional memory and throughput savings. On Llama-3.1-8B-Instruct, the measured CoCurve path lowers peak memory from 17.18 to 11.53 GB and raises prefill throughput to $1.41\times$ at 40%, while the full cross-model paths are given in Figure 12. Interaction-aware selection therefore expands the useful pruning-only region; common recovery and low-bit weights subsequently extend the same selected structures beyond its bend (Appendix D.2).

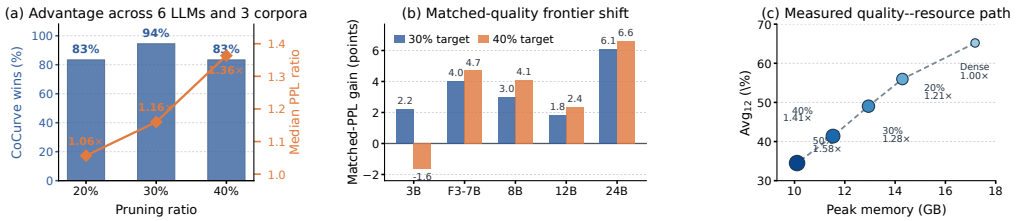

Figure 10: **Where CoCurve moves the measured deployment frontier. (a)** Win rate and median perplexity ratio against the strongest available baseline over 6 LLMs and 3 corpora at each shared operating point. **(b)** Matched-quality pruning-ratio shift on the five-model grid; positive values mean that CoCurve permits more pruning at the same WikiText-2 perplexity. **(c)** The measured Llama-3.1-8B-Instruct path in Avg_{12}, isolated-A100 peak memory, and prefill speedup. Values use linear interpolation in log-perplexity between measured baseline-envelope points without extrapolation.

D.2 Recovery and Quantization

D.2.1 Common-LoRA Recovery

We apply one common LoRA stage to the physically sliced 8B and 24B models from all five reported pruning methods, keeping the selected structures fixed. Table 12 gives the complete recipe: one tokenizer-packed Alpaca pass gives 589/601 optimizer steps for the 8B/24B models at an effective batch of 16 sequences.

Table 16: **Common-LoRA recovery by task.** Panel (a) reports full task sweeps on Llama-3.1-8B-Instruct and Mistral-Small-24B; panel (b) reports PPL-only CoCurve recovery on the remaining scales. Dense and CoCurve without LoRA are reference states; every recovered checkpoint uses the identical recipe in Table 12.

(a) Full two-model comparison.

		Perplexity ↓			Downstream accuracy ↑												
ρ	Method	Wiki	PTB	C4	PIQA	HS	WG	CSQA	A-e	A-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
								Llama-3.1-8B-Instruct									
0%	**Dense**	6.4	10.4	10.7	80.9	77.4	66.2	59.2	78.1	54.4	47.6	85.0	68.8	54.1	62.7	47.8	65.2±0.34
20%	Random	11.3	16.2	17.9	76.3	65.6	**62.0**	**50.7**	**68.6**	42.3	43.2	78.6	57.0	43.7	55.7	38.4	56.8±0.35
	LLM-Pruner	**9.9**	**14.7**	16.9	76.0	68.0	60.9	42.8	62.8	40.4	39.2	**81.2**	60.4	46.6	55.4	39.0	56.0±0.34
	SlimGPT	10.9	18.9	18.2	76.5	66.5	61.4	44.7	67.6	43.4	42.0	75.1	56.0	42.5	53.9	38.2	55.6±0.35
	SlimLLM	10.2	16.6	17.7	**76.7**	67.3	61.7	48.2	66.7	42.6	**43.6**	74.5	**61.1**	45.8	55.8	38.2	56.8±0.35
	CoCurve (w/o LoRA)	12.9	29.6	19.6	75.8	67.5	61.0	50.4	62.5	41.9	39.6	78.3	58.1	44.0	54.1	38.5	56.0±0.35
	CoCurve	10.0	18.6	**16.7**	76.4	**69.3**	60.5	45.1	65.3	**44.3**	42.0	79.7	60.2	**46.6**	**56.9**	**39.4**	**57.1**±0.35
30%	Random	15.3	21.4	22.7	71.8	57.1	58.3	42.7	60.7	37.3	**40.2**	72.5	53.6	40.5	53.3	34.5	51.9±0.35
	LLM-Pruner	13.0	**18.5**	20.8	72.4	60.6	56.8	40.6	56.6	35.4	37.8	64.9	57.2	**43.2**	54.7	35.2	51.3±0.35
	SlimGPT	14.7	26.1	23.7	73.1	57.6	58.6	**44.8**	61.5	**38.9**	37.2	72.9	52.5	39.1	51.3	34.6	51.8±0.35
	SlimLLM	13.9	22.9	22.5	**73.7**	60.0	57.7	43.2	**63.4**	37.3	38.6	63.4	54.0	41.0	53.1	35.0	51.7±0.35
	CoCurve (w/o LoRA)	21.6	36.3	32.0	70.5	55.8	55.5	41.0	55.6	34.4	35.2	73.1	46.5	37.3	50.1	33.4	49.0±0.35
	CoCurve	**12.6**	19.3	**20.8**	73.4	**62.2**	**59.7**	41.4	60.5	**38.9**	**40.2**	**76.4**	**57.6**	42.0	**55.6**	**36.2**	**53.7**±0.35
40%	Random	20.0	29.9	29.2	70.5	49.3	**56.7**	36.6	54.4	31.3	36.2	**68.0**	48.1	37.4	50.5	31.0	47.5±0.34
	LLM-Pruner	17.4	25.5	27.0	71.3	53.0	55.7	38.5	53.0	33.4	33.8	63.1	52.3	37.9	51.6	31.7	47.9±0.34
	SlimGPT	24.3	35.9	32.6	67.8	46.9	54.2	36.4	53.0	32.2	34.4	63.4	46.4	33.8	45.1	30.5	45.3±0.34
	SlimLLM	19.6	30.4	29.1	70.3	50.7	54.5	40.0	56.6	32.9	35.6	62.8	49.2	36.8	48.1	31.6	47.4±0.35
	CoCurve (w/o LoRA)	59.1	108.4	76.1	64.0	41.0	51.4	31.9	41.6	28.7	29.6	62.3	41.9	32.2	42.8	29.3	41.4±0.34
	CoCurve	**16.3**	**24.0**	**25.9**	**72.3**	**53.8**	56.3	**42.2**	**56.7**	**34.6**	**38.0**	64.2	**52.4**	**39.2**	**52.9**	**32.9**	**49.6**±0.35

continued on next page

Table 16(a), continued

ρ	Method	Perplexity ↓ Wiki	PTB	C4	Downstream accuracy ↑ PIQA	HS	WG	CSQA	A-e	A-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
50%	Random	32.1	44.5	39.8	65.7	40.3	**54.9**	32.7	48.7	25.8	33.2	62.1	45.5	34.1	46.1	29.4	43.2±0.34
	LLM-Pruner	28.0	38.3	37.1	**68.4**	45.0	53.8	35.8	47.3	**30.8**	32.2	62.3	45.5	34.9	48.3	**30.7**	44.6±0.34
	SlimGPT	41.8	60.6	50.9	64.3	36.3	53.1	31.2	47.4	25.9	30.0	61.5	38.9	30.3	41.4	28.4	40.7±0.34
	SlimLLM	31.1	45.8	40.9	66.0	41.0	49.9	34.4	48.9	27.8	29.4	62.3	41.8	32.0	41.3	29.0	42.0±0.34
	CoCurve (w/o LoRA)	956.4	1804.4	828.8	55.7	30.0	50.4	24.2	30.7	26.5	24.8	60.8	31.1	27.8	24.5	27.5	34.5±0.32
	CoCurve	**24.2**	**37.6**	**34.3**	68.3	**45.2**	53.4	**35.9**	**50.3**	29.5	**34.2**	**63.1**	**48.5**	**35.0**	**49.6**	30.3	**45.3**±0.34
	Mistral-Small-24B																
0%	**Dense**	4.7	24.6	8.5	83.0	81.6	75.1	54.5	82.6	59.6	48.0	86.8	68.4	51.0	57.3	52.3	66.7±0.33
20%	Random	6.7	23.4	11.9	79.5	76.8	68.5	53.2	78.7	55.6	46.8	86.5	68.2	51.3	59.7	47.1	64.3±0.34
	LLM-Pruner	6.8	**20.1**	11.8	80.0	77.6	**72.5**	54.4	**79.7**	58.3	**49.2**	86.6	**68.7**	**51.8**	59.0	47.8	**65.5**±0.34
	SlimGPT	6.7	25.3	11.7	80.3	78.2	69.5	54.5	77.7	55.8	47.0	84.0	66.7	50.1	57.4	47.1	64.0±0.34
	SlimLLM	**6.4**	23.5	**11.2**	**80.5**	**79.0**	71.7	51.8	78.5	**58.4**	48.4	**87.2**	67.4	51.3	**60.1**	**48.2**	65.2±0.34
	CoCurve (w/o LoRA)	9.0	131.2	14.9	77.8	76.0	66.3	51.2	74.1	52.7	42.0	79.7	63.6	44.3	55.9	43.5	60.6±0.34
	CoCurve	6.6	26.2	11.4	79.9	78.2	68.7	**55.1**	77.8	56.6	48.0	86.2	67.2	50.2	58.9	46.5	64.4±0.34
30%	Random	8.2	34.3	14.0	77.3	72.8	65.7	50.5	75.7	50.6	43.2	83.2	**65.5**	47.7	57.3	43.1	61.0±0.34
	LLM-Pruner	8.3	**28.5**	13.9	78.2	74.1	66.9	51.2	**77.0**	**54.4**	45.6	84.1	64.8	**48.9**	**57.8**	**44.3**	**62.3**±0.34
	SlimGPT	8.0	37.9	14.0	**79.3**	72.1	63.7	50.9	75.0	50.7	45.6	79.1	63.9	46.4	57.5	42.1	60.5±0.34
	SlimLLM	8.5	39.3	13.9	79.2	**74.7**	63.4	**51.9**	75.9	51.5	45.4	83.5	65.0	48.9	56.8	42.8	61.6±0.34
	CoCurve (w/o LoRA)	20.5	419.1	29.7	72.0	63.8	58.8	42.4	63.8	42.2	38.4	69.9	55.2	40.0	51.4	36.5	52.9±0.35
	CoCurve	**8.0**	33.0	**13.6**	78.2	73.7	**67.5**	49.7	74.7	51.8	**48.0**	**85.2**	64.1	47.4	57.7	42.7	61.7±0.34
40%	Random	11.5	47.1	17.1	76.0	67.4	61.5	49.1	70.4	44.5	40.2	80.9	60.4	44.5	53.6	**39.3**	57.3±0.34
	LLM-Pruner	10.4	**37.1**	16.6	74.9	68.5	62.9	49.0	71.9	47.0	43.8	80.9	61.8	45.4	**56.6**	39.2	58.5±0.35
	SlimGPT	10.1	52.2	17.2	75.8	65.8	57.1	47.2	70.7	45.6	40.2	69.3	59.0	42.9	54.2	38.3	55.5±0.35
	SlimLLM	11.4	44.9	17.0	**76.5**	67.4	59.0	46.8	70.6	45.7	42.4	75.9	61.4	44.4	54.3	38.4	56.9±0.35
	CoCurve (w/o LoRA)	277.1	4037.4	231.6	56.0	39.9	54.1	27.2	41.0	30.1	28.2	51.3	40.5	33.1	42.2	30.0	39.5±0.34
	CoCurve	**9.7**	81.0	**16.1**	75.6	**69.0**	**65.0**	**50.2**	**72.1**	**47.6**	**44.2**	**81.1**	**62.0**	**45.6**	55.4	39.1	**58.9**±0.34
50%	Random	16.4	71.4	21.3	72.1	60.0	58.7	**47.7**	65.5	41.0	**37.6**	74.0	57.2	41.5	51.5	35.3	53.5±0.35
	LLM-Pruner	14.4	**64.3**	21.0	71.6	61.3	60.5	45.1	65.4	40.4	37.2	74.7	**59.2**	42.3	53.4	**35.5**	53.9±0.35
	SlimGPT	**13.4**	65.4	22.2	71.8	56.0	54.4	41.9	62.0	38.6	35.8	67.2	53.3	39.4	50.3	33.3	50.3±0.35
	SlimLLM	15.6	70.6	22.1	**72.4**	58.5	54.7	42.9	65.2	41.0	**37.6**	69.3	54.7	39.3	50.0	33.8	51.6±0.35
	CoCurve (w/o LoRA)	2325.3	9329.9	1874.0	52.6	30.1	48.8	22.4	33.2	24.9	24.0	46.6	34.1	29.2	24.2	28.2	33.2±0.32
	CoCurve	14.7	134.1	**20.4**	71.1	**62.5**	**60.9**	41.9	**66.5**	**41.3**	37.4	**78.1**	59.0	**43.7**	**54.1**	34.8	**54.3**±0.34

(b) CoCurve PPL recovery across the remaining scales.

ρ	Llama-3.2-3B w/o LoRA W	P	C	+LoRA W	P	C	Falcon3-7B w/o LoRA W	P	C	+LoRA W	P	C	Mistral-Nemo-12B w/o LoRA W	P	C	+LoRA W	P	C	Llama-3.1-70B-It w/o LoRA W	P	C	+LoRA W	P	C
0%	6.9	10.5	10.4	–	–	–	5.4	9.0	12.2	–	–	–	5.0	35.2	8.9	–	–	–	3.7	8.0	8.5	–	–	–
20%	16.7	27.3	23.2	13.8	21.1	19.3	7.1	11.7	15.6	6.8	11.2	15.0	8.6	83.4	14.3	7.5	24.6	12.8	6.4	9.9	12.4	5.9	9.5	11.5
30%	34.8	49.4	55.0	18.9	25.4	27.5	8.9	15.1	19.5	7.8	13.1	17.1	15.8	160	23.3	9.5	34.3	15.5	8.4	13.6	17.2	6.9	10.7	13.2
40%	194	257	196	34.5	43.5	36.0	13.8	23.1	29.2	9.6	16.4	20.5	52.2	455	71.0	12.7	51.5	19.3	33.3	39.8	58.1	9.4	13.6	16.5
50%	4.6k	6.2k	3.4k	38.0	52.1	44.7	41.7	61.7	67.0	12.6	20.3	25.6	839	2.6k	643	18.0	81.3	25.9	713	1.7k	792	13.2	17.9	20.3

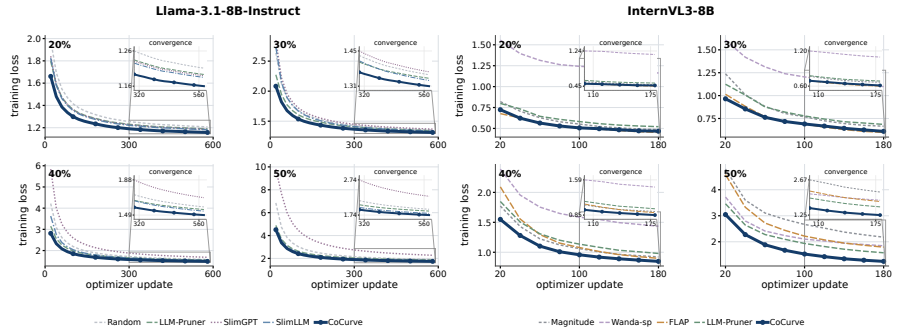

Figure 11: **Optimization trajectories under the common LoRA recipe.** The left 2×2 matrix reports Llama-3.1-8B-Instruct with an identical 589-update recipe; the right reports InternVL3-8B with an identical 188-update recipe. CoCurve has the lowest recorded LLM loss throughout and finishes top-two on every VLM trajectory, leading at 40/50%. Held-out results in Table 16 and Table 21, rather than training loss, establish recovered quality.

Table 3 gives the compact WikiText-2 and Avg_{12} view. Table 16 resolves the two full-evaluation models into all three perplexity corpora and all twelve downstream tasks across 20–50% pruning. Its second panel extends the CoCurve-only PPL comparison to the remaining model scales; it is used only to measure recovery, not to make downstream claims. At 70B, the same data, rank, effective batch, and optimizer schedule are applied as QLoRA on the experiment's NF4 base; the already evaluated CoCurve structure remains a fixed runtime mask because packed low-bit matrices cannot be sliced as ordinary dense tensors. Each panel includes the same CoCurve mask before recovery, exposing both the remaining gap to the original model and the improvement supplied by the common LoRA stage. Across all eight 8B/24B blocks, CoCurve is top-two in 14/16 WikiText-2/Avg_{12} columns. Across all six scales, the identical recipe lowers CoCurve's WikiText-2 perplexity

at all 24 measured model–ratio points. The effect remains substantial at the hardest completed boundary: at 50% on the 8B model, perplexity falls from 956.4 to 24.2 and Avg_{12} rises from 34.5 to 45.3. Thus recovery is not compensating for a single favorable checkpoint; it repeatedly exposes useful capacity retained by the selected structure.

Across the language panels, Figure 11 shows a lower training objective throughout the identical schedule, with particularly clear early separation at 30% and 40%. The VLM panels retain the same favorable ordering at the harder ratios. The held-out tables complete the argument: CoCurve's structures are easier to optimize under common recipes and retain better capability after optimization.

D.2.2 Composition with Quantization

We load the same fixed masks with bitsandbytes NF4 and INT8 weights (Table 12). Quality evaluation quantizes at loading and then applies each fixed mask; the systems columns independently time the corresponding physically sliced, low-bit checkpoint under the isolated A100 protocol. Dense NF4/INT8 isolate numerical compression, CoCurve bf16 isolates structural compression, and CoCurve+NF4/INT8 realizes their composition.

Across the two ratios, two precisions, and three corpora, CoCurve leads all 12 matched perplexity comparisons. Its four Avg_{12} entries are either best or within 0.2 points of the best alternative. Across the same CoCurve structures, NF4 halves bf16 peak memory while retaining 96% of prefill throughput, and physical slicing then moves beyond dense NF4 to lower-memory, higher-throughput operating points. Table 4 gives the compact quality–systems view, while Table 17 resolves all twelve downstream tasks.

Table 17: **Quantized pruning by task** on Llama-3.1-8B-Instruct. The same fixed masks are evaluated under NF4 and INT8 at 30% and 40% pruning.

			Perplexity ↓			Downstream accuracy ↑											
ρ	Method	Prec.	Wiki	PTB	C4	PIQA	HS	WG	CSQA	A-e	A-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
–	**Dense**	INT8	6.5	10.4	10.9	81.1	77.3	66.2	59.0	78.0	54.1	48.4	84.9	68.5	53.9	62.5	47.7	65.1
		NF4	6.8	10.8	11.5	79.9	76.9	66.4	59.8	76.2	52.8	46.2	85.4	67.0	53.0	61.9	46.7	64.3
30%	LLM-Pruner	INT8	26.8	39.6	40.8	69.9	54.1	52.4	**42.2**	50.4	33.1	30.8	**73.8**	**48.3**	**38.7**	47.6	31.8	47.8±0.34
		NF4	29.2	42.2	44.3	68.8	52.9	53.7	**42.3**	51.0	31.4	29.2	71.0	**48.0**	**37.9**	47.7	31.4	47.1±0.34
	SlimLLM	INT8	80.0	102.1	115.3	68.9	37.4	52.2	35.7	52.7	28.6	28.2	62.3	36.2	28.9	36.7	30.5	41.5±0.34
		NF4	80.0	101.5	114.4	67.8	36.9	51.7	36.4	**53.1**	28.2	28.8	62.3	36.7	29.3	37.1	30.4	41.6±0.34
	CoCurve	INT8	**21.7**	**36.9**	**32.3**	**70.7**	**55.2**	**55.9**	40.4	**54.5**	**35.1**	**35.4**	72.8	46.7	37.0	**50.3**	**33.3**	**48.9**±0.35
		NF4	**23.1**	**38.5**	**34.8**	**69.7**	**54.1**	**54.7**	37.8	51.0	**31.7**	**32.6**	**72.7**	45.7	36.4	**48.7**	**32.4**	**47.3**±0.34
40%	LLM-Pruner	INT8	101.3	162.9	124.9	**65.3**	**42.3**	**53.2**	**32.8**	40.3	27.8	**28.2**	**65.3**	37.0	**32.3**	40.1	**29.6**	**41.2**±0.34
		NF4	116.1	194.2	138.8	**64.8**	**42.2**	52.4	**33.9**	39.2	**27.7**	25.6	61.8	36.4	**31.3**	39.4	**29.1**	**40.3**±0.33
	SlimLLM	INT8	187.8	219.6	222.5	61.5	30.9	50.2	28.8	39.8	23.6	25.0	53.1	32.7	27.0	32.9	27.7	36.1±0.33
		NF4	192.8	272.0	242.6	61.4	30.1	**53.7**	29.1	**40.7**	23.6	25.4	49.5	32.0	27.0	33.2	27.7	36.1±0.33
	CoCurve	INT8	**57.6**	**105.4**	**75.0**	63.9	40.6	50.2	30.1	**41.2**	**29.9**	28.0	62.2	**41.2**	32.2	**43.2**	29.2	41.0±0.34
		NF4	**66.9**	**140.8**	**86.7**	62.2	39.8	50.9	30.2	38.1	27.4	**28.4**	**62.1**	**40.1**	31.1	**41.9**	28.6	40.1±0.34

D.3 Inference Efficiency

Table 18: **Inference efficiency of every timed method at** 20% **and** 40%**.** Llama-3.1-8B-Instruct, physical removal, one isolated A100-80GB.

	$\rho = 20\%$				$\rho = 40\%$			
Method	**Red.↑**	**Pref.↑**	**Dec.↑**	**Peak↓**	**Red.↑**	**Pref.↑**	**Dec.↑**	**Peak↓**
Dense	0.00	1.000	1.000	17.18	0.00	1.000	1.000	17.18
Random	17.44	1.196	1.006	14.31	34.89	1.462	1.021	11.44
Magnitude	17.41	1.167	1.011	14.38	34.89	1.389	1.038	11.58
Wanda-sp	17.41	1.224	0.996	14.25	34.82	1.468	1.009	11.45
FLAP	17.45	1.231	0.976	14.23	34.86	1.478	1.028	11.43
LLM-Pruner	17.44	1.196	1.008	14.31	34.84	1.430	1.031	11.51
SlimGPT	17.45	1.233	0.989	14.22	34.77	1.483	0.990	11.43
SlimLLM	17.52	1.219	1.021	14.26	34.83	1.517	1.010	11.37
CoCurve	17.40	1.211	1.020	14.29	34.81	1.415	0.998	11.53

Note. Param. is parameter reduction (%); Pref./Dec. are matched-dense throughput ratios; peak is generation GB.

Table 5 reports absolute throughput and peak memory, alongside relative gains from physical slicing. At fixed 20% and 40% budgets on Llama-3.1-8B-Instruct, the matched-method table separates se-

lection quality from system effects by timing every physically sliced mask under the same prompts, repeats, and isolated A100 protocol (Table 18). Decode remains sensitive to the selected widths, so the parameter budget alone does not determine kernel efficiency; the isolated measurements provide the quantitative systems comparison. Figure 12 extends the same trade-off view across model scales.

At each pruning ratio, all methods realize essentially the same parameter reduction and occupy narrow systems bands: prefill ranges from 1.167–$1.233\times$ at 20% and 1.389–$1.517\times$ at 40%, while peak memory spans only 14.22–14.38 and 11.37–11.58 GB, respectively. The quality ordering is therefore not explained by a more favorable physical footprint for CoCurve; selection improves which structure is retained, while slicing converts any such structure into the measured deployment gain.

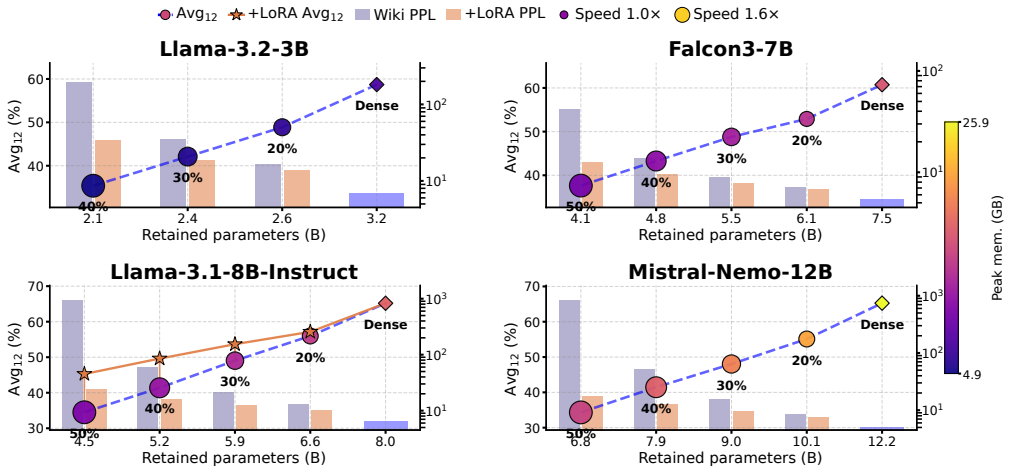

Figure 12: **Cross-model quality–efficiency frontiers.** The recovery stars appear only for the 8B model, whose post-recovery Avg_{12} was measured; every panel reports before/after-recovery perplexity.

Selection cost and amortization. Complete selection cost includes the reusable graph build, exact breakpoint enumeration, and held-out scoring of every distinct candidate mask; Table 19 reports all three rather than graph construction alone. For the 8B model, the five-ratio sweep evaluates $1{,}362$ distinct candidates and takes 17.4 recorded end-to-end hours, or 3.5 hours per compiled operating point after amortization. The same $\mathbf{H}$ serves all ratios and interaction strengths, and only a 2.3 MB Gram matrix is retained. Moreover, 768 independent single-unit sweeps replace the $294{,}528$ pairwise interventions of a direct pair-ablation design; this counts model evaluations rather than Gram arithmetic. The sweeps can be sharded without synchronization, reducing wall-clock but not aggregate compute. We document native pipeline cost rather than rank methods by wall-clock because published implementations combine different amounts of scoring, reconstruction, and recovery. The quality side of this trade-off is measured at the same operating points in Figure 3 and Table 1: on the headline 8B model the reusable build supports the 19.0% and 42.8% WikiText reductions at 30% and 40% over the strongest matched baseline.

Table 19: **Complete CoCurve selection-cost accounting over the 10–50% sweep.** Path time includes breakpoint-guided CPU enumeration and GPU scoring of each distinct mask on 16 held-out sequences; build time includes the 128-sequence single-unit feature pass and Gram construction. The 3B/8B runs use one RTX 6000 Ada and the packed-NF4 70B run one A100-80GB, so the measurements document cost rather than a cross-device speed ranking.

Model	Units	Build (h)	Candidates	Paths (h)	Total (h)	Stored H
Llama-3.2-3B	672	6.0	1,310	2.3	8.3	1.8 MB
Llama-3.1-8B-Instruct	768	12.6	1,362	4.8	17.4	2.3 MB
Llama-3.1-70B-Instruct (NF4)	1,920	43.1	2,361	19.7	62.8	15 MB

D.4 VISION–LANGUAGE RESULTS

D.4.1 PRUNING-ONLY TRANSFER

Table 20 resolves all seven tasks for three models at 20/30% pruning. Within each ratio block, the adjacent "diagonal only" and full CoCurve rows isolate the off-diagonal contribution while preserving the identical model, data, guard, and budget. Appendix D.4.2 separately tests whether one identical LoRA stage preserves recoverability at these operating points.

Table 20: **Complete vision–language results by benchmark.** Three representative models at 20% and 30%; ablations are excluded from method ranking.

ρ	Method	Caption PPL↓	Downstream accuracy ↑							Avg_7	Ret.
			MMB	SEED	SQA	AI2D	MMStar	POPE	MME		
	Qwen2.5-VL-7B										
	Dense reference	20.53	86.7	70.5	88.5	84.5	63.8	89.6	87.3	81.6±0.45	
20%	Random	16.34	71.4	61.2	70.1	61.9	43.5	80.1	62.3	64.4±0.56	65%
	Magnitude	**14.51**	79.9	65.1	73.1	69.9	53.6	87.3	**81.7**	72.9±0.52	83%
	Wanda-sp	17.43	73.8	56.7	72.0	63.5	43.2	74.4	56.5	62.9±0.56	62%
	FLAP	15.17	80.4	63.4	78.6	70.5	50.6	**88.6**	80.9	73.3±0.51	83%
	LLM-Pruner	19.22	80.5	61.9	**78.8**	73.6	52.5	84.7	73.5	72.2±0.52	81%
	ShortGPT (layer drop)	433.69	31.5	28.9	35.6	24.3	26.6	51.0	49.0	35.3±0.56	6%
	CoCurve (diagonal only, λ=0)	16.27	81.1	66.3	75.4	70.6	53.9	87.6	79.4	73.5±0.51	84%
	CoCurve	17.96	**81.9**	**67.1**	77.8	**75.2**	51.5	86.9	80.5	**74.4**±0.50	86%
30%	Random	24.24	58.7	44.3	46.1	33.8	37.6	52.1	50.7	46.2±0.59	28%
	Magnitude	17.94	65.2	54.9	56.0	39.5	38.4	84.4	**72.2**	58.7±0.56	54%
	Wanda-sp	19.21	63.4	44.3	54.0	44.6	33.3	71.3	54.2	52.2±0.58	41%
	FLAP	**15.91**	65.6	50.5	58.2	45.1	35.7	83.2	71.5	58.5±0.56	53%
	LLM-Pruner	18.12	61.5	43.4	**59.9**	43.6	37.2	82.3	**72.2**	57.2±0.56	51%
	ShortGPT (layer drop)	3811.02	32.5	29.5	35.7	23.0	25.6	51.0	49.1	35.2±0.56	6%
	CoCurve (diagonal only, λ=0)	17.48	67.7	55.8	55.5	52.6	40.6	84.9	70.5	61.1±0.56	59%
	CoCurve	19.88	**68.1**	**59.7**	59.5	**55.3**	**41.5**	**90.7**	70.1	**63.6**±0.55	64%
	Qwen3-VL-8B										
	Dense reference	19.41	89.7	69.4	94.4	84.8	66.9	90.4	88.4	83.4±0.43	
20%	Random	19.26	59.7	44.5	48.2	34.8	35.6	85.0	68.7	53.8±0.56	42%
	Magnitude	15.82	68.4	57.3	62.6	57.9	43.3	58.0	51.2	57.0±0.59	48%
	Wanda-sp	26.49	38.3	33.8	46.0	33.7	27.6	52.2	51.8	40.5±0.58	16%
	FLAP	15.54	74.9	61.4	69.8	62.4	47.6	88.5	**74.5**	68.4±0.54	71%
	LLM-Pruner	19.12	74.3	**63.1**	74.5	60.4	46.5	81.7	60.4	65.8±0.55	66%
	ShortGPT (layer drop)	9093.50	30.1	26.1	38.1	26.2	26.0	48.8	51.1	35.2±0.56	6%
	CoCurve (diagonal only, λ=0)	13.43	72.8	59.1	68.4	59.7	45.4	88.7	71.6	66.5±0.54	67%
	CoCurve	18.10	**81.1**	61.8	**76.3**	**69.2**	**52.4**	**88.7**	68.0	**71.1**±0.53	76%
30%	Random	35.15	41.6	36.5	36.9	28.3	30.4	63.0	51.9	41.2±0.57	18%
	Magnitude	55.45	34.9	29.8	35.0	24.5	27.5	59.6	52.4	37.7±0.56	11%
	Wanda-sp	48.60	34.5	29.6	36.1	23.8	27.5	49.0	50.8	35.9±0.56	7%
	FLAP	**24.95**	42.2	41.3	42.4	28.8	31.4	82.5	57.0	46.5±0.56	28%
	LLM-Pruner	30.17	43.4	47.1	39.5	25.0	29.5	**85.4**	59.0	47.0±0.55	29%
	ShortGPT (layer drop)	41895.50	33.2	27.4	34.8	23.1	28.1	49.5	50.8	35.3±0.56	6%
	CoCurve (diagonal only, λ=0)	25.43	49.2	43.7	46.5	33.1	33.7	83.3	61.2	50.1±0.57	35%
	CoCurve	25.40	**62.6**	**53.2**	**54.2**	**47.3**	**39.3**	83.9	**63.7**	**57.7**±0.57	50%
	InternVL3-8B										
	Dense reference	14.05	88.2	67.4	97.5	83.7	67.1	93.3	88.1	83.6±0.42	
20%	Random	14.68	72.9	61.0	74.3	62.9	49.7	87.0	67.6	67.9±0.54	69%
	Magnitude	**13.46**	79.8	57.7	74.6	64.1	48.3	91.1	79.3	70.7±0.52	75%
	Wanda-sp	19.17	50.0	36.3	62.6	47.0	31.9	65.8	55.1	49.8±0.58	34%
	FLAP	14.72	80.7	62.1	**84.8**	**73.8**	52.6	88.3	81.2	74.8±0.50	83%
	LLM-Pruner	14.53	78.4	61.8	78.7	68.4	52.2	84.0	79.0	71.8±0.52	77%
	ShortGPT (layer drop)	20.33	33.8	32.2	61.7	51.5	24.1	51.0	49.2	43.4±0.57	22%
	CoCurve (diagonal only, λ=0)	15.04	81.0	62.0	77.9	71.0	53.3	91.0	80.8	73.9±0.51	81%
	CoCurve	14.04	**83.7**	**64.2**	82.7	72.3	**53.8**	**92.5**	**82.3**	**75.9**±0.49	85%
30%	Random	19.48	58.9	45.0	48.7	38.6	34.9	62.5	57.9	49.5±0.59	34%
	Magnitude	17.41	47.5	42.2	46.0	32.1	30.2	65.8	52.3	45.2±0.58	25%
	Wanda-sp	21.74	49.2	36.3	51.5	34.8	28.6	56.5	58.3	45.0±0.58	25%
	FLAP	18.27	**74.5**	**56.1**	**65.0**	**54.7**	**42.1**	83.6	53.1	61.3±0.56	57%
	LLM-Pruner	19.31	65.3	53.2	61.4	52.5	40.4	84.8	64.7	60.3±0.56	55%
	ShortGPT (layer drop)	26.04	31.8	29.5	37.8	26.3	25.7	49.7	48.9	35.7±0.56	7%
	CoCurve (diagonal only, λ=0)	17.30	62.2	49.1	54.0	50.3	36.0	80.0	56.9	55.5±0.57	45%
	CoCurve	**16.64**	67.7	54.0	59.6	52.7	41.6	**89.0**	**71.9**	**62.4**±0.55	59%

Across all six model–ratio blocks, full CoCurve has the highest Avg_7 among the pruning methods. Relative to the strongest external baseline, the margin is 1.1–2.7 points at 20% and 1.1–10.7 at 30%; relative to its diagonal-only counterpart, the edge term adds 0.9–7.6 points across the full grid. The consistent ordering across three VLM architectures shows that the interaction signal transfers beyond text-only models, while the growing 30% margins identify where modeling those interactions matters most.

D.4.2 RECOVERY WITH A COMMON TRAINING RECIPE

We recover fixed Qwen2.5-VL-7B, Qwen3-VL-8B, and InternVL3-8B masks from 20% through 50% with one recipe for all methods and the same seven-benchmark harness. Across the resulting 12

model–ratio blocks, CoCurve ranks first or second on Avg_7 in 11, including the best 50% results on Qwen2.5-VL-7B and InternVL3-8B and second place on Qwen3-VL-8B (Table 21). The common training stage therefore extends the pruning-only advantage through 50%, showing that structure selection and recovery remain complementary under heavier compression. MMB, SEED, SQA, and MMStar denote MMBench, SEED-Bench, ScienceQA, and MMStar; Avg_7 weights the seven benchmarks equally.

Table 21: **VLM recovery by benchmark.** Caption perplexity and seven downstream tasks under the common LoRA recipe.

ρ	+LoRA method	PPL↓	Downstream accuracy ↑							Avg_7
			MMB	SEED	SQA	AI2D	MMStar	POPE	MME	
Qwen2.5-VL-7B **Dense:** PPL 20.53; Avg_7 81.6										
20%	Magnitude	**11.9**	80.7	67.5	74.0	71.5	**55.5**	**90.0**	**86.3**	75.1±0.50
	Wanda-sp	15.2	75.5	59.1	71.0	65.3	48.9	88.3	76.6	69.2±0.53
	FLAP	12.8	80.9	65.5	**80.1**	73.3	53.6	89.2	84.6	75.3±0.50
	LLM-Pruner	15.4	81.0	65.1	77.9	**75.1**	54.9	88.6	83.0	75.1±0.50
	CoCurve	14.9	**83.0**	**68.9**	78.4	75.0	54.2	89.3	84.6	**76.2**±0.49
30%	Magnitude	14.9	75.0	62.8	66.5	60.2	48.3	90.4	81.3	69.2±0.53
	Wanda-sp	15.3	69.5	56.5	61.8	49.5	40.5	82.7	69.4	61.4±0.56
	FLAP	**14.2**	73.3	64.0	70.7	58.6	44.4	88.4	81.3	68.7±0.53
	LLM-Pruner	15.7	**77.6**	61.6	**73.8**	**63.5**	**49.8**	88.4	**81.5**	**70.9**±0.52
	CoCurve	16.4	76.0	**64.8**	65.6	61.5	47.3	**92.1**	80.8	69.7±0.52
40%	Magnitude	19.8	68.1	58.2	57.5	45.6	42.4	86.0	74.0	61.7±0.56
	Wanda-sp	20.1	55.9	47.7	49.6	36.2	30.1	76.3	56.4	50.3±0.57
	FLAP	**17.8**	65.1	57.8	58.9	48.0	39.8	86.9	73.7	61.5±0.56
	LLM-Pruner	20.2	**72.5**	**59.0**	**60.4**	**50.2**	**44.4**	82.3	61.3	61.4±0.56
	CoCurve	19.9	68.0	56.9	58.4	47.8	36.7	**90.1**	**75.4**	**61.9**±0.55
50%	Magnitude	64.9	57.6	40.8	48.1	31.3	**36.2**	65.2	59.2	48.3±0.58
	Wanda-sp	**45.5**	35.9	27.2	34.6	26.7	24.1	52.2	50.1	35.8±0.56
	FLAP	63.6	39.7	37.4	43.3	31.5	28.4	63.9	53.9	42.6±0.57
	LLM-Pruner	46.4	57.9	46.5	47.6	33.6	32.2	77.0	56.0	50.1±0.57
	CoCurve	60.5	**62.1**	**51.9**	**52.8**	**38.8**	36.0	**81.9**	**61.9**	**55.1**±0.57
Qwen3-VL-8B **Dense:** PPL 19.41; Avg_7 83.4										
20%	Magnitude	14.7	77.8	60.3	69.7	62.3	47.8	89.5	74.3	68.8±0.53
	Wanda-sp	17.7	52.5	42.0	63.2	47.9	34.1	64.8	54.5	51.3±0.58
	FLAP	**13.8**	77.2	63.6	72.9	66.3	51.1	88.7	75.4	70.7±0.53
	LLM-Pruner	15.8	79.3	**64.9**	79.1	66.8	52.5	89.9	77.0	72.8±0.51
	CoCurve	15.8	**83.2**	63.9	**79.6**	**70.6**	**54.8**	**93.2**	**80.4**	**75.1**±0.50
30%	Magnitude	25.2	66.4	53.2	55.8	45.8	40.1	83.4	63.4	58.3±0.57
	Wanda-sp	32.3	37.4	32.4	51.3	35.8	28.2	51.0	49.8	40.8±0.58
	FLAP	**20.8**	67.5	58.4	60.5	52.4	41.9	91.7	74.7	63.9±0.55
	LLM-Pruner	23.6	74.6	**63.3**	67.1	58.5	**50.4**	89.6	**77.0**	68.6±0.54
	CoCurve	22.6	**77.8**	61.7	**72.0**	**63.5**	48.6	**92.2**	71.0	**69.5**±0.53
40%	Magnitude	90.9	32.3	22.5	35.0	27.1	27.0	52.6	52.7	35.6±0.56
	Wanda-sp	80.1	33.7	24.8	36.2	27.2	27.2	51.0	49.2	35.6±0.56
	FLAP	**31.0**	61.8	52.0	48.6	43.7	35.0	83.2	57.4	54.5±0.57
	LLM-Pruner	55.8	**67.2**	**55.8**	**57.3**	**46.2**	**43.4**	**87.7**	**61.3**	**59.8**±0.56
	CoCurve	38.0	60.4	44.9	52.4	34.6	32.8	80.2	59.3	52.1±0.57
50%	Magnitude	1.1e6	27.9	22.0	35.6	25.9	21.7	51.0	49.2	33.3±0.55
	Wanda-sp	1.5e3	28.6	23.7	36.7	**27.8**	21.8	51.0	48.6	34.0±0.55
	FLAP	**207.1**	29.3	22.5	36.1	27.3	22.7	51.0	49.2	34.0±0.55
	LLM-Pruner	524.9	**36.1**	**29.2**	**38.7**	23.8	**27.3**	**51.5**	49.5	**36.6**±0.56
	CoCurve	2.1e3	31.4	28.1	34.6	27.5	22.1	50.0	**50.1**	34.8±0.56
InternVL3-8B **Dense:** PPL 14.05; Avg_7 83.6										
20%	Magnitude	**12.2**	82.1	63.2	80.4	70.3	54.4	92.7	81.9	75.0±0.50
	Wanda-sp	15.7	51.7	41.4	63.3	50.3	32.1	59.9	58.2	51.0±0.58
	FLAP	12.9	80.9	65.8	**85.4**	**74.4**	53.8	92.4	83.3	76.6±0.48
	LLM-Pruner	13.1	80.8	64.7	81.4	73.0	**56.1**	91.3	83.4	75.8±0.49
	CoCurve	12.8	**83.5**	**66.0**	83.6	73.8	54.5	**92.8**	**84.1**	**76.9**±0.48
30%	Magnitude	**14.8**	73.2	57.1	65.0	51.7	40.7	90.4	74.0	64.6±0.54
	Wanda-sp	16.7	56.7	43.8	55.8	44.8	31.2	58.6	56.6	49.6±0.59
	FLAP	16.9	**76.6**	**61.6**	**72.1**	61.5	47.9	**91.9**	77.4	**69.9**±0.53
	LLM-Pruner	16.9	71.7	60.0	70.0	**63.9**	**50.9**	88.7	75.6	68.7±0.54
	CoCurve	15.2	75.7	60.1	69.4	59.7	48.7	91.0	**80.1**	69.2±0.53
40%	Magnitude	24.5	62.8	52.0	52.3	40.3	34.7	**87.2**	65.5	56.4±0.56
	Wanda-sp	27.1	41.5	36.5	46.7	38.8	29.1	66.7	55.0	44.9±0.58
	FLAP	26.1	67.0	**53.3**	**57.1**	47.7	36.9	60.6	59.3	54.6±0.59
	LLM-Pruner	25.3	62.5	52.8	55.4	41.4	**39.1**	71.9	62.4	55.1±0.58
	CoCurve	**22.3**	**68.5**	51.4	55.4	**49.7**	37.8	82.7	**71.0**	**59.5**±0.56
50%	Magnitude	4.3e3	28.0	22.7	34.9	27.5	23.5	51.0	49.2	33.8±0.55
	Wanda-sp	67.2	29.8	27.1	35.1	27.9	26.1	51.1	49.2	35.2±0.56
	FLAP	234.4	27.9	22.1	35.2	28.2	21.0	51.5	50.4	33.8±0.55
	LLM-Pruner	71.2	43.3	37.1	40.2	29.3	31.2	53.8	56.7	41.7±0.58
	CoCurve	**51.0**	**61.7**	**47.2**	**48.8**	**38.4**	**35.5**	**83.4**	**61.9**	**53.8**±0.57

InternVL3-8B optimization trajectories are shown beside the LLM trajectories in Figure 11; held-out VLM results remain the criterion in Table 21.

D.4.3 VLM Quality–Efficiency Frontier

Figure 13 combines selection-only and recovered quality with isolated-A100 measurements after physical slicing, exposing the VLM quality, throughput, and memory trade-off under one protocol. At 50% pruning, prefill throughput rises by 1.22–1.45× and peak memory falls by 42–43% across the three architectures; these systems gains accompany rather than replace the recovered-quality comparison.

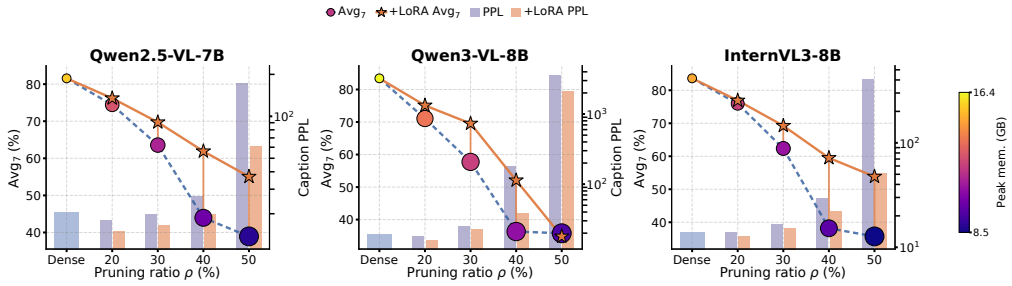

Figure 13: **VLM quality–efficiency frontiers.** Each panel follows one model from dense through 50% pruning, combining Avg_7, caption perplexity before and after common recovery, prefill speedup, and isolated-A100 peak memory.

D.5 Additional Behavioral Checks

D.5.1 Fidelity to the Dense Model

Because selection minimizes token-level KL to the dense model, agreement with dense predictions is its direct behavioral test; downstream accuracy asks whether that fidelity is useful. We measure the fraction of shared items on which the pruned and dense models choose the same normalized answer. Across 20–40% pruning, CoCurve leads all twelve capability-group comparisons and all three Avg_{12} columns, while ranking first or second on 35 of 36 individual task comparisons. Table 22 therefore tests the objective at the decision level across the main operating range without introducing another benchmark or calibration signal.

Table 22: **Agreement with the dense model's prediction** (%, Llama-3.1-8B-Instruct). Columns report all twelve tasks in Appendix C.1; Avg_{12} weights them equally.

ρ	Method	PIQA	HS	WG	CSQA	A-e	A-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
20%	Random	80.1	69.9	66.7	63.2	61.8	54.9	61.8	67.8	63.4	63.8	69.4	58.5	65.1
	Magnitude	78.6	71.0	66.5	63.2	63.4	55.3	60.2	66.8	65.4	65.3	70.2	58.0	65.3
	Wanda-sp	85.0	74.2	70.4	73.7	74.3	66.2	**72.0**	74.4	70.3	67.3	69.2	65.1	71.8
	FLAP	85.0	75.9	69.0	70.0	74.4	64.9	71.6	72.6	66.8	65.4	65.9	64.5	70.5
	LLM-Pruner	86.1	81.3	73.6	73.8	71.2	65.7	68.6	83.4	**75.9**	**73.7**	76.3	66.7	74.7
	SlimGPT	85.3	75.7	67.5	70.2	73.8	64.2	70.8	65.4	64.5	62.7	66.5	62.8	69.1
	SlimLLM	86.6	77.3	71.0	72.8	**77.1**	66.9	70.6	70.0	69.7	68.9	71.5	66.8	72.4
	CoCurve	**88.2**	**83.5**	**74.7**	**76.5**	75.1	**69.1**	71.8	**83.5**	75.2	73.4	**78.9**	**69.2**	**76.6**
30%	Random	70.8	53.9	59.7	51.7	50.5	44.9	52.4	48.0	56.0	56.5	60.7	47.0	54.3
	Magnitude	62.5	42.0	54.5	42.6	39.1	37.5	43.4	52.5	42.0	46.3	48.1	41.3	46.0
	Wanda-sp	76.4	52.8	62.4	56.3	55.7	51.4	56.6	46.8	55.6	54.1	57.2	51.7	56.4
	FLAP	74.1	55.1	60.7	55.9	53.5	47.3	57.2	51.9	49.9	52.3	51.8	49.8	55.0
	LLM-Pruner	80.3	68.9	**67.6**	61.8	59.7	55.5	58.4	**75.1**	**62.7**	**64.5**	67.6	55.4	64.8
	SlimGPT	69.6	52.8	60.4	49.5	50.0	44.0	53.6	50.2	46.6	50.8	53.5	47.5	52.4
	SlimLLM	80.5	53.6	64.5	60.7	63.5	54.3	60.8	63.2	50.6	52.2	55.0	55.0	59.5
	CoCurve	**81.0**	**71.3**	67.2	**64.5**	**64.8**	**58.9**	**62.2**	75.0	62.4	62.9	**71.0**	**58.0**	**66.6**
40%	Random	59.8	38.4	54.0	39.7	37.2	34.6	41.4	43.0	41.1	42.3	46.7	40.4	43.2
	Magnitude	54.4	28.8	49.9	24.2	28.6	28.8	35.8	48.7	28.0	29.2	29.1	28.5	34.5
	Wanda-sp	59.9	33.2	50.4	37.2	35.4	34.9	37.6	37.8	30.4	33.8	29.2	38.4	38.2
	FLAP	55.7	32.3	50.9	32.4	31.6	30.3	33.2	41.3	28.3	30.1	25.9	31.9	35.3
	LLM-Pruner	71.8	**53.2**	55.7	45.9	45.4	40.4	44.0	**65.7**	43.6	47.3	53.8	45.2	51.0
	SlimGPT	53.2	30.1	50.3	24.9	31.1	29.7	31.6	37.3	28.4	31.8	22.2	35.0	33.8
	SlimLLM	70.5	44.3	**58.6**	49.5	**51.1**	44.5	49.4	50.1	43.1	44.9	45.3	46.9	49.8
	CoCurve	**72.3**	52.8	54.1	**50.5**	49.7	**46.6**	**52.0**	63.1	**54.1**	**53.9**	**58.1**	**48.8**	**54.7**

Its Avg_{12} agreement margin over the strongest baseline is 1.9, 1.8, and 3.7 points at 20%, 30%, and 40%, respectively. The objective's token-level fidelity therefore survives at the decision level and remains useful across capability groups, even though no downstream benchmark enters selection.

D.5.2 Few-Shot Prompting

Five-shot prompting puts subject-matched demonstrations before each MMLU question, changing the inference protocol while keeping the evaluated knowledge benchmark fixed. The experiment asks whether the ordering found under zero-shot evaluation survives when the pruned model must use information supplied in context. We treat it as a protocol-robustness test and do not aggregate it into the zero-shot Avg_{12}.

Table 23: **Five-shot MMLU** (%). Subject-dev demonstrations are identical across pruning methods; dense references are gray. All five available model–ratio comparisons favor CoCurve.

		Zero-shot			Five-shot		
Model	ρ	**LLM-P**	**SlimLLM**	**CoCurve**	**LLM-P**	**SlimLLM**	**CoCurve**
Llama-3.1-8B-Instruct; Dense zero/five=47.8/50.8							
	20%	36.8	37.2	**38.5**	34.9	34.8	**37.0**
	30%	31.9	30.4	**33.4**	31.3	27.0	**32.7**
	40%	**29.7**	27.6	29.3	28.7	26.7	**28.9**
Llama-3.2-3B; Dense zero/five=41.8/41.5							
	20%	33.5	32.0	**33.9**	33.0	31.0	**33.6**
	30%	29.5	28.8	**30.0**	29.3	28.4	**30.1**

Table 23 uses demonstrations drawn from each subject's dev split and identical for every method. The ordering is the one the zero-shot tables show: CoCurve leads the strongest evaluated baseline in all five comparisons by 0.2–2.1 points. The advantage the selection buys is therefore not confined to the zero-shot prompt format.

D.6 Comparison with a Native Smaller Model

This comparison asks whether pruning can retain inherited capacity from a larger pretrained model and outperform a dense native-smaller checkpoint under a lower peak-memory budget. Table 24 holds the evaluation harness fixed and contrasts a dense bf16 parent, the recovered and quantized CoCurve checkpoint, and a native smaller checkpoint in both precisions. For model pairs with different tokenizers, the matched claim uses the shared twelve-task suite rather than cross-model perplexity.

Table 24: **Full comparison with a native smaller model.** The main comparison is summarized in Table 2.

Role	**Checkpoint**	**Prec.**	**Load↓**	**Peak↓**	**Wiki↓**	**Cmn_4↑**	**Sci_3↑**	**Read_4↑**	**Know_1↑**	**Avg_{12}↑**
Native-small memory–capability comparison										
Native parent	Mistral-Small-24B	bf16	23.57	45.02	4.7	73.5	63.4	65.9	52.3	66.7±0.33
Native small	Mistral-7B	bf16	7.25	14.16	4.8	69.5	57.4	61.9	44.1	61.8±0.34
Native small	Mistral-7B	NF4	7.25	4.51	4.9	69.6	57.0	61.9	43.9	61.7±0.34
CoCurve	Mistral-Small-24B, 20%	NF4	19.11	**12.14**	6.8	**69.6**	**59.5**	**64.9**	**45.5**	**63.5**±0.34

Note. Load: B parameters; Peak: maximum allocated GB in the fixed forward workload.

All rows use the same task harness, and all memory rows use the same isolated-A100 measurement protocol. The recovered 20% CoCurve checkpoint uses less peak memory than native Mistral-7B in bf16 while improving every capability group and Avg_{12} by 1.7 points. The dense bf16 parent anchors inherited quality, while the native-small bf16/NF4 rows separate structural inheritance from numerical compression.

E What the Co-Pruning Edges Reveal

We ask a simple interpretability question: what structure does the co-pruning graph expose, and when does that structure matter for pruning? This part answers it through one mechanism chain. Appendix E.1 moves from graph structure to conditional decisions and matched interventions; Appendix E.2 isolates the edge term from eligibility and granularity; Appendix E.3 tests whether the quadratic ranks fixed-budget masks; and Appendix E.4 asks whether the resulting graph and decisions survive estimator and calibration perturbations. Together, the sections connect the object CoCurve estimates to the choices it changes and the behavior those choices preserve.

E.1 From Graph Structure to Decisions and Behavior

Figure 14 shows that the off-diagonal is structured by both depth and module pairing rather than behaving like homogeneous estimation noise. Viewed as an output-space co-pruning atlas, the graph locates units whose individual perturbations point in similar directions, including relations that cross both layer and module boundaries. This organization motivates examining units by their aggregate connectivity and then testing whether that relational structure changes actual selections.

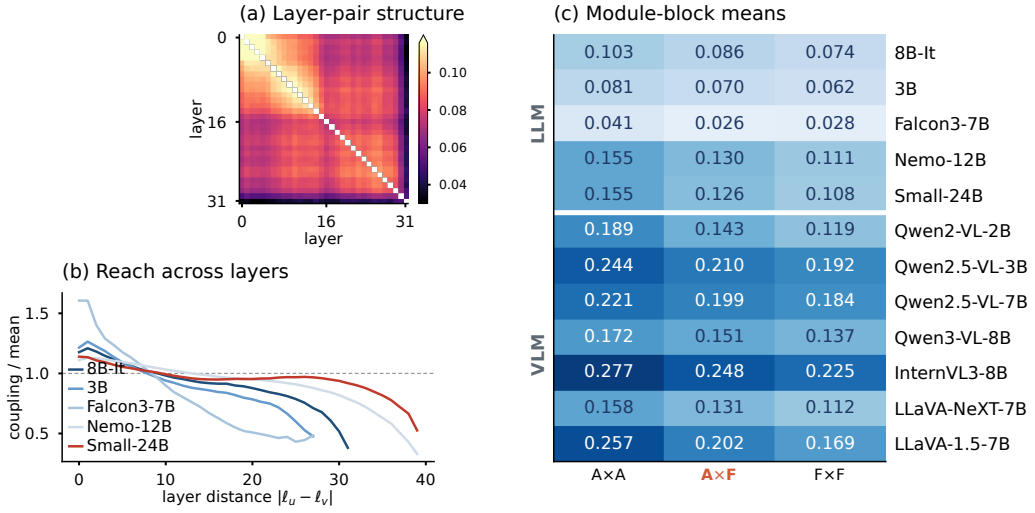

Figure 14: **Extended anatomy of the estimated interaction matrix.** (a) Layer-pair means on Llama-3.1-8B-Instruct; (b) correlation by layer distance relative to the all-pair mean; (c) module-block means across language and vision–language models. Within-module means exclude self-correlations. Near-layer coupling is strongest, yet the slower cross-depth decay and consistently large attention–FFN block (highlighted in orange) show that the estimated graph is neither local-only nor separable by module.

Bridge units. With $d_u = \mathbf{H}_{uu}$ and $b_u = \sum_{v \neq u} |\mathbf{H}_{uv}|$, the fixed operational definition is $d_u < Q_{0.40}(d)$ and $b_u > Q_{0.60}(b)$ on every model. Whether CoCurve or the diagonal selector retains those units is an outcome, not part of the definition. The intersection occupies only 3.5–14.5% of the inventory across all ten language-model matrices (Table 28), making it a selective relational population rather than a broad low-saliency partition. The first test is consequently decision-level: at an identical greedy prefix, does accumulated coupling change which eligible unit is least costly to remove?

A concrete conditional decision. Table 25 uses the current, cost-corrected Mistral-Nemo-12B mask. All three displayed units are eligible at step 186. The selected path point reproduces the deployed mask exactly. The diagonal and calibrated masks remove 188 and 187 units, respectively, because attention and FFN costs differ; both satisfy the realized-ratio check. The calibrated mask retains attention unit 458 and FFN unit 472 in layer 19 and removes FFN unit 593 in layer 24 instead. Its selection-split calibration-risk advantage is 3.1 paired standard errors. Thus the trace identifies

Table 25: **A real edge-induced decision on Mistral-Nemo-12B.** Three eligible candidates at step 186 of the deployed 20% selection. Scores are cost-normalized and multiplied by 10^{12}; layer indices are zero-based. All three satisfy the same guard.

Unit	Layer	Individual	Accumulated edge	Conditional total
Attention group 458	19	**237.005**	4.926	241.932
FFN group 472	19	239.280	2.458	241.738
FFN group 593	24	239.289	2.385	**241.674**

an actual conditional ordering change, rather than inferring one from marginal saliency or from the final mask alone.

Causal consequence of bridge structure. Design. We test the bridge population on the five models in the full LLM comparison. The forced population uses the full unit inventory, distinct from the guarded selector's eligible-candidate analysis. On each, we force-prune the bridge units (low d_u, high b_u) and, as a control, an equal-size set drawn from the same low-saliency pool but with low connectivity, matched unit-for-unit on saliency d_u, unit type and per-unit cost. The remainder of the budget is one cost-normalized greedy fill, computed once and shared by both arms. The two masks that result hold the same number of units, the same attention/FFN split, the same removed cost and the same realized ratio to the last bit, and differ in which low-saliency population was forced out. On Llama-3.1-8B-Instruct that is 169 units each, 1.399×10^9 parameters, realized ratio 0.2005, 82 units shared and 87 forced.

Removing the high-connectivity units is far more destructive on capability (Table 26): nineteen of the twenty capability cells fall, by up to 23.7 points, despite the two forced populations being matched on individual saliency, unit type, and cost. The four groups show that the effect is distributed across behavior rather than carried by one benchmark family.

Table 26: **Matched bridge intervention by task** across five language models (ρ=20%). Bridge/control sets match saliency, unit type, and cost and share one greedy fill; Δ = Bridge − Control, so negative values indicate greater damage from bridge removal. Avg_{12} weights all tasks equally.

Model	Forced set	PIQA	HS	WG	CSQA	ARC-e	ARC-c	OBQA	BoolQ	R-m	R-h	QuAIL	MMLU	Avg_{12}
Llama-3.2-3B	Control	72.5	60.8	54.5	46.3	57.3	36.5	35.0	61.1	43.2	36.8	41.8	35.2	48.4
	Bridge	67.8	41.9	51.2	34.2	51.1	28.2	31.4	57.0	38.8	30.4	39.9	30.7	41.9
	Δ	−4.7	−18.9	−3.2	−12.0	−6.3	−8.3	−3.6	−4.1	−4.4	−6.4	−1.9	−4.5	−6.5
Falcon3-7B	Control	75.1	63.8	57.5	44.6	69.0	43.1	34.2	69.9	51.6	39.3	48.7	38.4	52.9
	Bridge	71.1	60.8	55.4	35.9	62.8	38.1	29.2	70.7	53.1	38.7	50.0	37.1	50.2
	Δ	−4.1	−3.0	−2.1	−8.7	−6.1	−4.9	−5.0	+0.9	+1.5	−0.6	+1.3	−1.3	−2.7
Llama-3.1-8B-It	Control	77.1	70.9	62.7	55.3	70.3	45.7	43.0	52.3	57.5	46.3	52.7	40.5	56.2
	Bridge	73.4	53.3	52.2	35.6	52.6	31.1	33.2	64.6	43.5	32.5	41.9	32.0	45.5
	Δ	−3.7	−17.6	−10.5	−19.7	−17.7	−14.7	−9.8	+12.3	−14.0	−13.7	−10.8	−8.5	−10.7
Mistral-Nemo-12B	Control	77.9	72.5	60.1	47.7	69.3	47.6	38.4	72.1	58.8	45.4	52.1	40.1	56.8
	Bridge	73.6	51.0	52.6	35.2	52.4	32.2	33.4	50.7	40.9	32.5	41.8	31.7	44.0
	Δ	−4.2	−21.4	−7.5	−12.4	−16.9	−15.4	−5.0	−21.4	−17.9	−12.9	−10.3	−8.4	−12.8
Mistral-Small-24B	Control	72.9	73.1	62.3	47.4	71.1	46.1	37.8	77.6	56.5	43.4	51.9	40.7	56.7
	Bridge	56.4	29.7	50.4	27.4	39.6	22.0	26.2	38.0	33.8	28.1	34.5	25.9	34.3
	Δ	−16.5	−43.5	−11.9	−20.1	−31.5	−24.1	−11.6	−39.6	−22.6	−15.2	−17.4	−14.8	−22.4

The five-model intervention deliberately uses the full bridge population, but bridge prevalence varies with depth (Table 28). We therefore repeat the comparison with each forced bridge matched to its nearest-saliency low-coupling control in the same layer, unit type, and parameter-cost stratum. Only strata containing both populations contribute a pair; the remainder of the budget is again one shared fill. Across both models, bridge removal raises independent test KL and WikiText perplexity and lowers Avg_{12} (Table 27). The effect is modest but consistent on 3B and substantially larger on 8B, showing that the bridge signal is not reducible to a coarse depth preference.

Layer structure and allocation. Table 28 verifies the bridge prevalence quoted in the main text and shows that aggregate coupling is systematically strongest in earlier layers across all ten models, whereas diagonal saliency has no consistent depth direction. The edge matrix therefore contributes a relational depth signal that cannot be recovered by applying one monotone depth correction to individual scores.

Table 27: **Depth-matched bridge intervention** at ρ=20%. The matching above additionally fixes layer; both arms share one greedy fill. Positive Δ uniformly indicates greater damage from bridge removal; Test-KL uncertainty is the paired sequence-level standard error over 128 independent sequences.

Model	Pairs	$\lvert\Delta d\rvert$ ($\times 10^{-4}$)	Δ Test KL ↑	Δ Wiki ↑	Δ Avg$_{12}$ loss ↑
Llama-3.2-3B	35/77	4.81	**0.023**$_{\pm 0.006}$	**1.05**	**0.44**
Llama-3.1-8B-It	42/81	2.69	**0.090**$_{\pm 0.005}$	**0.92**	**2.04**

Table 28: **Depth dependence of individual saliency and aggregate coupling.** Bridge is the percentage of units satisfying the fixed low-saliency/high-coupling definition; the remaining columns report Spearman rank correlation between layer index ℓ_u and individual saliency $\mathbf{H}_{uu}$ or total coupling $b_u = \sum_{v \neq u} |\mathbf{H}_{uv}|$. Each row uses the same $\mathbf{H}$ build evaluated in Table 33. Coupling is negative on all ten; the saliency column changes sign across models.

Model	L	Bridge (%)	$\rho_s(\ell_u, \mathbf{H}_{uu})$	$\rho_s(\ell_u, b_u)$
Llama-3.2-3B	28	12.8	−0.082	**−0.874**
Falcon3-7B	28	5.4	+0.143	**−0.543**
Mistral-7B	32	14.5	+0.107	**−0.880**
Qwen2.5-7B	28	5.0	−0.115	**−0.878**
Llama-3.1-8B-It	32	11.3	−0.229	**−0.867**
Yi-1.5-9B	48	10.3	+0.331	**−0.806**
Falcon3-10B	40	8.0	+0.031	**−0.745**
Mistral-Nemo-12B	40	12.0	−0.225	**−0.910**
Llama-2-13B	40	3.5	−0.384	**−0.884**
Qwen2.5-14B	48	13.7	+0.403	**−0.821**
Range / median $\lvert\rho_s\rvert$		3.5–14.5	0.184	**0.871**

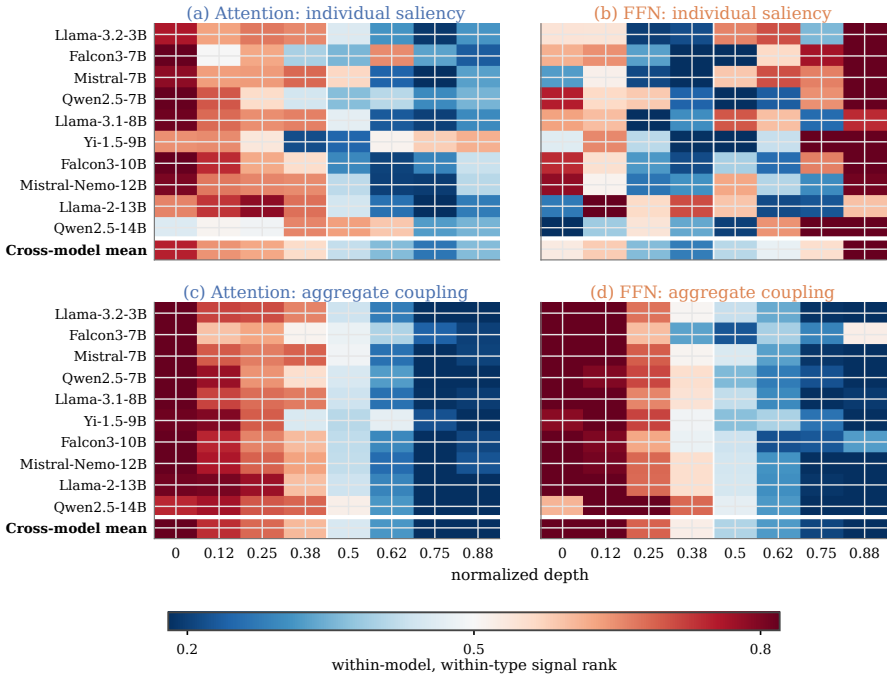

Figure 15: **Where node and edge signals concentrate across depth.** Each cell averages a within-model, within-unit-type percentile rank in one normalized-depth bin, so color compares spatial organization without conflating raw scales across architectures. Individual saliency is architecture- and module-dependent in **(a,b)**; aggregate coupling instead concentrates toward earlier depth for both attention and FFN in **(c,d)**. The separated bottom row is the cross-model mean; the matrices visualize the rank correlations in Table 28.

The contrast in Figure 15 is not a generic instruction to protect all early units: the solver uses signed conditional costs, and individual selections remain heterogeneous within each depth band. It instead identifies a stable source of information unavailable to a node-only selector—early units tend to participate in more consequential co-removal relations even when their isolated saliency is unexceptional.

Allocation. Figure 16 gives the per-layer, per-type profile on four models. On Llama-3.1-8B-Instruct at ρ=20% the solution removes 46 attention units and 105 FFN groups over 26 of 32 layers, at most nine in any one layer, and splits evenly over the second half of the depth alone, 46 attention against 46 FFN. The shared budget thus learns a heterogeneous allocation without collapsing onto one layer or module family; the depth pattern emerges from conditional costs rather than from a hand-set per-type schedule.

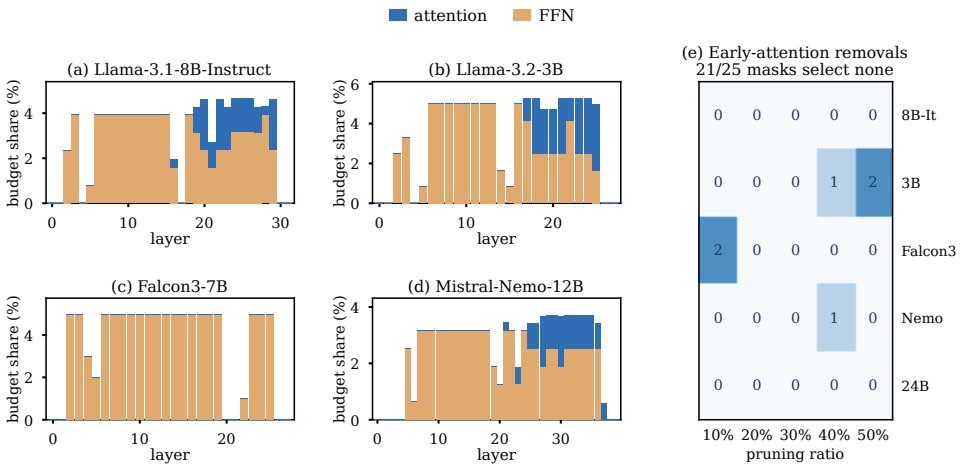

Figure 16: **Per-layer allocation of the deployed masks. (a)–(d)** Each layer's attention/FFN contribution to the shared removed-parameter budget at ρ=20%; this cost-weighted view remains comparable when unit sizes differ across architectures. **(e)** Attention units removed from the first half of depth over five models and five ratios: 21/25 deployed solutions remove none, and the largest exception removes two. The guard constrains per-layer totals but does not impose this allocation.

E.2 What the Edge Term Contributes

Table 29: **Effective contribution of co-pruning edges across six language models.** Δ is excess KL relative to the calibrated mask on the *selection split*; σ is the paired mean gap divided by its standard error on that same split. Neither is an independent significance test. The fixed-allocation block holds the diagonal mask's removed cost of each type fixed.

		No edges		**Undamped edges**		**Fixed split**	
Model	$\lvert\mathcal{F}\rvert$	Δ%	σ	Δ%	σ	Δ%	σ
Llama-3.2-3B	288	+9.6	3.3	+12.4	3.8	+11.5	4.2
Falcon3-7B	181	+6.6	8.2	+29.7	23.1	+7.6	11.4
Mistral-7B	318	+5.8	5.9	+32.0	11.0	+5.4	4.6
Llama-3.1-8B-It	303	+6.7	4.6	+13.0	6.0	+8.5	5.4
Yi-1.5-9B	400	+3.8	2.1	+746.9	39.1	+4.4	3.1
Falcon3-10B	402	+2.5	3.9	+30.1	8.3	+3.6	3.5

Across the six models in Table 29, removing edges raises held-out risk by 2.5–9.6% and changes the selected mask by Jaccard distance 0.13–0.53 (median 0.22), while fully undamping them is worse on every model. On the independent split, adding edges while holding the diagonal selector's allocation fixed improves test KL and WikiText-2 perplexity in all four evaluated blocks (Table 30). The fixed-split and free-allocation arms need not order identically on next-token KL and a heterogeneous downstream mixture; neither uses downstream labels for selection. Together, the controls isolate

Table 30: **Independent evaluation behind Table 7.** Test KL uses 128 sequences disjoint from graph estimation and λ selection; its small type is the sequence-level standard error. The fixed-split arm preserves the diagonal selector's attention/FFN allocation, so its improvement isolates within-type reordering by edges.

Model	ρ	Selection rule	Test KL ↓	Wiki PPL ↓	Avg_{12} ↑
Llama-3.2-3B	20%	Diagonal	$0.791_{\pm 0.020}$	17.1	45.8
		Edges, fixed split	$\mathbf{0.734}_{\pm 0.020}$	16.9	48.5
		CoCurve	$0.754_{\pm 0.020}$	**16.7**	**48.9**
	30%	Diagonal	$1.709_{\pm 0.032}$	49.0	39.8
		Edges, fixed split	$\mathbf{1.382}_{\pm 0.025}$	36.6	41.0
		CoCurve	$1.577_{\pm 0.029}$	**34.8**	**42.1**
Llama-3.1-8B-It	20%	Diagonal	$0.611_{\pm 0.014}$	13.1	55.1
		Edges, fixed split	$\mathbf{0.571}_{\pm 0.014}$	**12.5**	54.9
		CoCurve	$0.584_{\pm 0.013}$	12.9	**56.0**
	30%	Diagonal	$1.169_{\pm 0.017}$	26.0	48.7
		Edges, fixed split	$\mathbf{1.016}_{\pm 0.017}$	22.0	**49.7**
		CoCurve	$1.085_{\pm 0.018}$	**21.6**	49.0

the robust source of the gain: relative to the diagonal selector, edges improve unit ordering beyond merely changing the attention/FFN split, while calibration prevents their local curvature signal from dominating the finite intervention.

The following controls isolate two structural design axes under the same budget and guard; estimator-budget sensitivity is treated separately in Appendix E.4.

(i) Module eligibility. Table 31 rules out a simpler interpretation in which the result follows merely

Table 31: **Restricting eligibility to one module type** at a 20% target, with the deployed guard and $\hat{\lambda}$. *attn/FFN* gives the removed-unit split; Falcon3-7B attention reaches its guarded capacity at 9.1%, while its joint solution chooses only FFN and therefore equals the FFN-only arm.

	Falcon3-7B				Llama-3.2-3B			
Eligible units	ρ	**attn/FFN**	**Wiki↓**	**Cmn_4 (%)↑**	ρ	**attn/FFN**	**Wiki↓**	**Cmn_4 (%)↑**
Attention only	9.1%	96/0	851	36.9	20.1%	180/0	465	37.7
FFN only	20.2%	0/101	**7.07**	**59.4**	20.1%	0/120	**16.42**	55.0
Both (**CoCurve**)	20.2%	0/101	**7.07**	**59.4**	20.1%	38/95	16.66	**56.6**

from choosing the right module family. On Llama-3.2-3B, where all arms reach the target, joint eligibility improves the capability trade-off over FFN-only pruning and avoids the severe perplexity loss of attention-only pruning. On Falcon3-7B, the selector instead spends the entire feasible budget on FFN, adapting the module split rather than enforcing one in advance.

(ii) Interaction granularity. The edges could act at a coarser level than unit pairs: keep the cross-component block but replace it with its scalar mean, and the interaction decides how much of each component type to remove and never which unit. That is the level SAViT (Zheng et al., 2022) interacts at, and Table 32 runs it. Because averaging the block rescales the edge mass that enters the marginal, that arm is calibrated at its own $\hat{\lambda}$, isolating interaction granularity from calibration. It still loses six of the seven columns, including all four capability groups; only PTB is better. What the edges contribute is therefore which *pairs* are costly together, not a per-component budget.

Table 32: **Interaction granularity** (Llama-3.1-8B-Instruct, ρ=20%). Both rows use the same solver, guard and budget and differ only in whether the off-diagonal acts per unit pair or per component pair; each is independently calibrated and realizes its target ratio to within 0.2 points.

Interaction granularity	WikiText↓	PTB↓	C4↓	Cmn.↑	Sci.↑	Read.↑	Know.↑
CoCurve (unit pair)	**12.88**	29.63	**19.58**	**63.7**	**48.0**	**58.6**	**38.5**
component pair	12.90	**28.81**	19.84	62.8	47.2	57.9	37.8

E.3 Does the Surrogate Rank Masks?

Same-budget discrimination. Correlation across ratios can conflate ranking quality with the changing budget, so we evaluate the harder case of one fixed budget. We fix ρ=20% on Llama-3.1-8B-Instruct and construct a pool of 26 masks at the same target budget: the CoCurve-greedy set, the diagonal-greedy set, 14 random sets and 10 local-swap perturbations of the CoCurve set, each filled to within 0.75% of the same removed cost. We compare predicted $\frac{1}{2}s^\top \mathbf{H}s$ against measured $D_{\mathrm{KL}}(p_0 \| p_S)$ across the pool (Figure 17). The surrogate reaches full-pool Spearman 0.87 and separates the groups completely: the twelve structured selections average D_{KL} 0.62 over 0.57–0.71, whereas every random set is worse, averaging 1.26. We then remove this between-group signal and test the twelve structured masks alone, all within 0.15 nats of one another. Spearman correlation remains 0.55, and reaches 0.58 among the ten local swaps; this fine-grained discrimination most closely matches the comparisons made by the greedy solver.

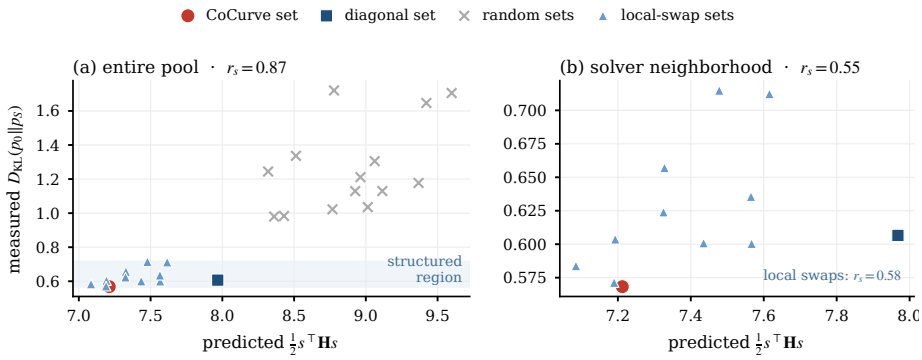

Figure 17: **The surrogate discriminates at one fixed budget.** Predicted $\frac{1}{2}s^\top \mathbf{H}s$ against measured $D_{\mathrm{KL}}(p_0 \| p_S)$ for masks matched to within 0.75% of the same removed cost on Llama-3.1-8B-Instruct at ρ=20%. **(a)** The full pool reaches Spearman 0.87 and places every structured set below every random set. **(b)** The structured-only view isolates fine-grained ordering and retains correlation 0.55 (0.58 among local swaps).

E.4 Estimator and Calibration Robustness

Three questions about stability, each answered by a different perturbation: resampling the held-out set, changing the selection protocol, and rebuilding the curvature from independent calibration draws.

Stability of the estimated matrix. We rebuild $\mathbf{H}$ independently on random halves of the calibration set, resampling over *sequences* (positions within a sequence are strongly dependent, so treating the stored positions as i.i.d. would understate the error). Across ten models spanning six families and 3B–14B the off-diagonal entries reproduce across halves at Pearson 0.885–0.996 and the diagonal at 0.994–1.000 (Table 33). The agreement supports reproducible interaction structure across calibration halves. The finite-ablation substitution and joint-additivity approximation behind the estimator are measured in Appendix B.2.

Sensitivity to the surrogate budget. The edge estimator has two finite-budget knobs: the calibration size (number of sampled positions) and the top-r logit support of the Fisher-Gram features. Neither choice drives the result. We rebuild $\mathbf{H}$ on the cheapest model (Llama-3.2-3B) at top-$r \in \{128, 256, 512\}$ and calibration size $\in \{64, 128\}$, then run each build through the same selector—the guard of Appendix C.2 and λ re-chosen on held-out KL for that build—and evaluate (Table 34).

Table 33: **Split-half stability of H**, resampled over calibration sequences, mean of four random halvings. Edges remain strongly reproducible across all ten models, while the diagonal terms are near-identical.

Model	Individual terms (Pearson)	Co-pruning edges (Pearson)
Llama-3.2-3B	0.996	0.885
Falcon3-7B	0.998	0.979
Mistral-7B	1.000	0.962
Qwen2.5-7B	1.000	0.991
Llama-3.1-8B-It	0.994	0.905
Yi-1.5-9B	1.000	0.966
Falcon3-10B	0.997	0.965
Mistral-Nemo-12B	0.999	0.982
Llama-2-13B	1.000	**0.996**
Qwen2.5-14B	0.999	0.950

Across the builds measured, WikiText-2 perplexity spans 3.1% and MMLU 0.4 points. The two larger top-r supports select the same 133-unit mask despite using separate feature builds, while the remaining settings preserve downstream quality with only small mask changes. This stability supports $(128, 256)$ as a reliable default surrogate budget. Figure 18 asks the same question of the selection directly rather than through a benchmark: how many calibration sequences are needed before the chosen units stop changing.

Table 34: **Surrogate-budget sensitivity** (Llama-3.2-3B at ρ=20%, under the deployed guard). Upper block varies top-r at fixed calibration 128; lower block varies calibration at fixed top-256. The default is $(128, 256)$. Each build re-estimates $\mathbf{H}$, recalibrates $\hat{\lambda}$, and reselects its mask.

Axis	Setting	Wiki↓	MMLU (%)↑
Top-r	128	17.17	34.0
Top-r	256 (default)	16.66	33.9
Top-r	512	16.66	33.9
Calibration sequences	64	16.82	33.6
Calibration sequences	128 (default)	16.66	33.9

Selection-split and protocol robustness. The primary LLM masks use one calibration draw and 16 held-out selection sequences. Two questions follow, and they are different: whether the *selection step* would survive a different draw of the held-out set, and whether the *curvature it selects over* would survive a different draw of the calibration text. We answer both.

Held-out resampling concentrates on a narrow candidate set. The path files record per-sequence risk at every candidate, so the bootstrap reuses these risks without additional model forwards. Over 26 paths spanning ten models and five ratios, with 2000 resamples each, bootstrap selections occupy a median of only 3.3% of the enumerated family, and 2.5% at the three main ratios. A typical family contains 318 candidate masks and the whole bootstrap moves among ten of them.

The criterion identifies a mask near the risk minimum rather than a uniquely meaningful numerical value of λ: evaluated back on the full held-out set, the mask chosen by a resample incurs a median of 0.00% additional risk relative to the deployed mask, and every path puts at least 98.5% of resamples within one standard error of the minimum. The solution is piecewise constant and the path is flat near its optimum, so the members the bootstrap moves among are the ones that cannot be told apart.

Downstream stability around the selected risk minimum. The bootstrap judges $\hat{\lambda}$ by held-out risk, the quantity it is selected on. The independent check is to take the selected point and its neighbors and evaluate all of them on the full suite. On Llama-3.2-3B at ρ=20% we do this for five points: $\hat{\lambda}$ itself, the two members at the edges of its 1-SE band, and the two nearest solutions just outside it. The selected point is within 0.3 points of the best of the five on three of the four capability

groups and within 0.91 on the fourth, and it is 2.7 to 2.9 points above the worst on the twelve-task mean. Thus the selected mask preserves favorable downstream quality within the risk region that the held-out sample cannot distinguish.

Changing the protocol. A bootstrap only perturbs the sample. We therefore varied the protocol itself on Llama-3.1-8B-Instruct at ρ=20%: replacing KL by NLL, increasing the held-out set from 16 to 48 sequences, and using two further draws all select the bit-identical deployed solution. Together with the split-half and budget tests above, this establishes stability at the graph, path, and deployed-decision levels.

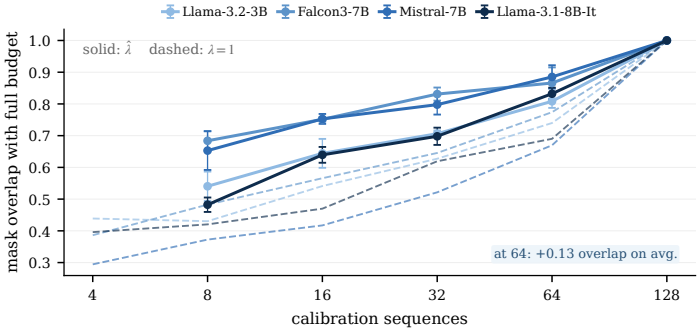

Figure 18: **How much calibration data the selection needs.** Overlap between the units chosen from n sequences and those chosen from the full budget of 128, mean over three draws. Measuring the selection rather than downstream accuracy separates estimator convergence from benchmark resolution. At 64 sequences, calibrated $\hat{\lambda}$ raises overlap by 0.13 on average relative to λ=1; damping the off-diagonal, the part of the matrix that needs the most data, also makes the solution easier to estimate.

F Discussion and Reproducibility

F.1 Extended Related Work

Signals for structured post-training pruning. Weight magnitude provides a data-free reference, while Wanda and FLAP incorporate activation statistics into weight or structure scores (Sun et al., 2024; An et al., 2024). LLM-Pruner couples related structures with first-order Taylor importance (Ma et al., 2023); SlimGPT uses layer-wise OBS reconstruction (Ling et al., 2024); and Slim-LLM combines head selection, FFN feature analysis, and regression repair (Guo et al., 2025). AMP and 2SSP focus explicitly on allocating sparsity across modules or stages (Mugnaini et al., 2025; Sandri et al., 2025). SubspacePath Pruner couples scenario representation subspaces to reusable head-wise pathways through lightweight probes, targeting inference-time specialization rather than task-agnostic joint attention–FFN co-pruning (Gong et al., 2026a). Adjacent compression families answer different contracts: SparseGPT and the original Wanda remove individual weights without changing dense tensor shapes; ShortGPT removes entire Transformer blocks; SliceGPT rotates the retained basis before width slicing; and LoRAPrune interleaves structural decisions with adaptation (Frantar & Alistarh, 2023; Sun et al., 2024; Men et al., 2025; Ashkboos et al., 2024; Zhang et al., 2024). They are important context but not matched pruning-only selectors over the shared attention-head/FFN-unit inventory used by our primary grid. These methods motivate broad signal and allocation coverage in Appendix C.3, but none makes a global predictive-risk edge between two prospective removals the quantity accumulated during selection.

Curvature, interactions, and forward perturbations. OBD scores isolated removals with a curvature diagonal (LeCun et al., 1989), whereas OBS and WoodFisher repair retained weights using curvature (Hassibi & Stork, 1992; Singh & Alistarh, 2020). CBS studies cancellation among jointly removed weights (Yu et al., 2022); SOSP-I constructs pairwise interactions between CNN structures (Nonnenmacher et al., 2022); and Kwon et al. retain within-layer Fisher interactions after head and filter counts have been fixed (Kwon et al., 2022). SAViT uses cross-component Taylor importance for vision Transformers (Zheng et al., 2022), and LLM-Surgeon combines Kronecker-factored curvature with weight repair (van der Ouderaa et al., 2024). Bonsai instead learns an additive quality predictor from forward perturbations (Kolawole et al., 2024). CoCurve joins the useful premises behind these lines: finite forward ablations define output-space features; their Gram matrix exposes pairwise attention–FFN interactions globally; and a conditional pruner accumulates the signed incident edges of the current removal set.

Recovery, numerical compression, and VLMs. LoRA can recover a fixed pruned structure (Ma et al., 2023; Ling et al., 2024), or be interleaved with structural pruning (Zhang et al., 2024). Separating recovery from selection lets the primary grid identify structure quality, while the common-recipe study measures recoverability. Likewise, NF4/INT8 composition asks whether structural and numerical compression coexist, not whether low-bit weights substitute for selecting a smaller computation graph. Efficient-VLM work predominantly reduces visual-token computation (Chen et al., 2024; Zhang et al., 2025; Cao et al., 2024; Zou et al., 2026); parameter-structural approaches use model-specific search or recovery (Shi et al., 2023; Liang et al., 2025). Our VLM experiment instead holds the predictive-risk formulation fixed and changes only the architecture-facing unit registry.

This taxonomy defines the matched comparison contract: structural selectors over one attention–FFN inventory are compared separately from repair, token pruning, and routed execution, whose nominal sparsity has a different operational meaning.

F.2 Scope, Limitations, and Extensions

The evidence establishes three operating conditions and points to direct extensions.

- **Upfront, reusable selection.** Building the interaction graph is costlier than an isolated-score pass, but it is performed once and reused across pruning ratios. The measured cost is therefore best amortized when several operating points are compiled or a checkpoint is deployed repeatedly (Appendix D.3).
- **Pairwise local model.** The graph is a second-order, pairwise approximation rather than an explicit model of all higher-order removals. Held-out coefficient selection, fixed-budget surrogate

tests, and the measured ratio sweeps validate this approximation over the reported operating range (Appendix E.3, Appendix E.4).

- **Architecture-facing deployment backend.** We instantiate CoCurve and physical slicing for attention and FFN units in dense language and vision–language Transformers. Routed experts or hybrid blocks require an appropriate unit registry and slicing backend, although the teacher-KL graph and conditional objective remain unchanged. They also expose a different loaded-versus-active parameter contract, so we do not present active-count matching as a pruning baseline.

These conditions concern the current implementation and measured regime; the co-pruning objective extends directly to broader structural inventories and deployment backends.

F.3 Broader Impact

Structured, training-free compression can lower the hardware and energy threshold for serving capable models. Physically smaller dense checkpoints are especially relevant where accelerator memory, cost, connectivity, or data-governance constraints prevent reliance on large hosted systems, widening the feasibility of local, institutional, and privacy-preserving deployment. Because CoCurve uses unlabeled calibration text without task-specific fine-tuning, one reusable graph can compile several deployment points, allowing practitioners to choose a quality–resource trade-off for their setting rather than inherit a single model size. These benefits do not make compressed models universally safe or equitable: they inherit the source model's limitations and may shift capabilities unevenly, so safety-, bias-, and domain-critical behavior should be reevaluated at the chosen operating point.

F.4 Assets, Licenses, and Ethics

All models are public pretrained checkpoints used under their respective licenses: Llama-2, Llama-3.1, and Llama-3.2 (Llama Community License); Falcon3 (Falcon LLM License); Mistral-7B, Mistral-Nemo, Mistral-Small, Yi-1.5, Qwen2.5-7B/14B, Qwen2.5-VL, and Qwen3-VL (Apache-2.0); and InternVL3 (MIT release with its Qwen2.5 component under the Qwen License). Evaluation datasets are used under their published terms. Baseline implementations follow the authors' released code where available and their published descriptions otherwise; Appendix C.3 records every deviation. No model is trained or fine-tuned in the pruning-only setting, and the recovery stage of Appendix D.2.1 uses the cleaned Alpaca corpus under its stated license.

F.5 Reproducibility Checklist

Method and configurations. The supplement specifies the structured-unit registry, runtime-mask and physical-slicing semantics, single-unit ablations, blockwise Gram estimator (Algorithm 1), conditional solver and guard (Algorithm 3), and breakpoint-guided interaction path. Appendix C records models, data splits, budgets, prompts, metrics, seeds, and recovery/deployment configurations. Selection uses no labels, gradients, or recovery, so a mask is determined by the checkpoint, calibration and selection splits, pruning ratio, and configuration.

Artifacts and verification. Code is included in the supplementary material, together with the estimator and solver, experiment configurations, pruning solutions, evaluation harness, and scripts that regenerate the reported tables from structured evaluation outputs. Physical slicing is checked against runtime masking before timing; FP32 equivalence checks isolate structural correctness, while bf16 drift and realized parameter counts are recorded. Optional language and VLM recovery follow the frozen recipes in Appendix C.6 and Appendix D.4.2; quantized conditions are marked explicitly rather than treated as the default precision. Each systems table states its accelerator, precision, batching, attention backend, and whether the machine was otherwise idle, separating portable parameter counts from hardware-dependent memory and throughput.