IN-CONTEXT LEARNING AS A REHEARSAL OF RECOVERY FOR DEPTH PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Depth pruning removes whole transformer blocks and then repairs the model. Existing criteria choose the blocks by the damage a removal does before repair. But they cannot tell which of two equally damaging removals repair will undo, because that depends on whether the remaining layers can still learn. Repairing and testing every candidate is out of reach (removing 9 of 36 layers alone leaves 94 million sets), so practice falls back on heuristics such as dropping a contiguous block of late layers. We instead find a better set with a few forward passes, drawing on the view that in-context learning is an implicit weight update. We therefore score a removal by how much it hurts the model's in-context learning ability. Over 1.6K prune-and-repair runs on five language models, we find that this probe helps predict post-repair quality better than present-damage scores alone, and more so with demonstrations than without, as our view expects. The same runs show why the choice matters. At a fixed budget, the set of removed layers changes post-repair quality far more than the choice among published criteria, and no single position is safe on every backbone. At 25% removal our method has the best five-model average under both repair objectives, even ahead of a baseline that briefly fine-tunes every candidate, for a few percent of pipeline cost. We also extend the study to vision–language models, where most prior depth pruning cuts only the language tower. We find that how the budget is split between the two towers decides which capabilities survive repair.

1 INTRODUCTION

Large language models are increasingly expected to run on phones, robots and other edge devices, where memory and latency are fixed by hardware chosen long before the model. A trained model must then be shrunk to fit. Depth pruning (or layer pruning), which removes whole transformer blocks, is the simplest way to do so in practice. It needs no architectural change, it reduces compute and memory together (Lu et al., 2026), and unlike unstructured sparsity or width pruning it speeds up generation on standard kernels (Song et al., 2024; Kim et al., 2024).

The pruned model is then repaired by fine-tuning (Kim et al., 2024; Gromov et al., 2025; Lu et al., 2026), and the repaired model is the one deployed. Yet almost every criterion for choosing the blocks scores the damage to the *unrepaired* model. We instead ask whether repair will be able to undo a removal. A line of work interprets in-context learning as an implicit weight update, in which demonstrations in the prompt change a model's predictions as gradient steps on them would (von Oswald et al., 2023; Dai et al., 2023; Akyürek et al., 2023). If so, a pruned model that can still learn a new rule from its prompt should be easy to repair. Testing this takes a few forward passes, about 3% of a prune-and-repair run (Appendix B).

Lu et al. (2026) argue that nothing needs to be measured, since, after fine-tuning, removing the last layers beats elaborate criteria. Across more than 1.6K prune-and-repair runs on five backbones, however, we find no fixed position that is safe on every model. The safe block lies deep on four backbones and shallow on the fifth (§4.2), and with budget, repair and backbone fixed, post-repair quality varies far more across removal sets than across criteria (Figure 1). Which layers to remove must therefore be measured. Repairing every candidate, as DarwinLM (Tang et al., 2026) does for width pruning, costs a training run each, and depth pruning has combinatorially many candidates.

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

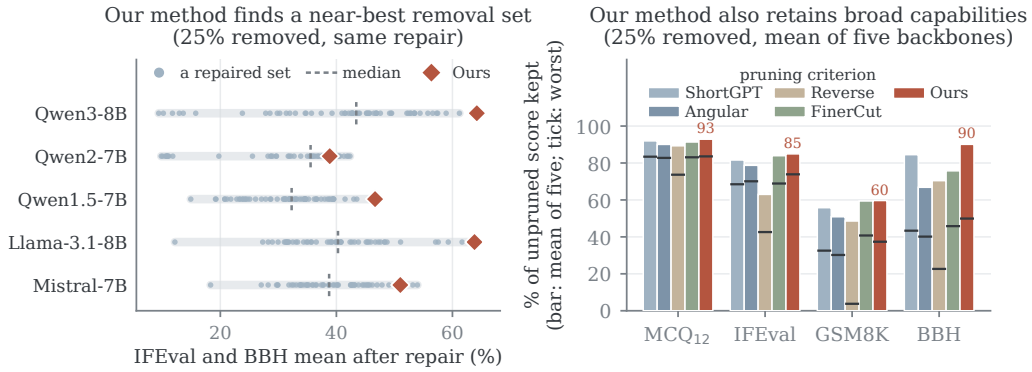


Figure 1: **Which layers go matters more than which published criterion picks them, and our probe finds sets that repair well.** All runs remove 25% of the layers and use the same repair. **Left**: post-repair quality of the 52–65 contiguous-block and random sets repaired per backbone (one run each), and of our selection (diamond, mean of three seeds). **Right**: share of the unpruned score kept after repair, five-backbone mean (bars) and worst backbone (ticks). Our method has the best mean on every axis. More details are in Appendix E.2 and Table 2.

Our proposal. We treat in-context learning as a rehearsal of the repair (Figure 2). A removal that breaks learning from the prompt should be hard to repair, and scores computed without demonstrations cannot see this difference (§3.1). Our probe therefore measures learnability, as a stand-in for recoverability, with in-context tasks that cannot be answered without their demonstrations. While the model reads the demonstrations, the probe compares its next-token distributions with and without a layer, by Jensen–Shannon divergence. It needs only forward passes, yet it raises the five-backbone average by about 2 points over the strongest baseline under either repair objective (§4.3).

Contributions.

- **Finding: a pruned model that can still learn from its prompt recovers better after repair.** Over 1.6K prune-and-repair runs on nearly 800 removal sets of five language models, the probe score of the removed layers predicts post-repair quality better than every present-damage score we measured, and better with demonstrations than without (§4.2).
- **Method: a layer selector that needs no training, only forward passes.** Any present-damage score can propose candidates (block influence by default), and our probe chooses among them (§3.2). At 25% removal it has the best five-backbone average under both repair objectives, ahead of our depth port of DarwinLM, which briefly fine-tunes every candidate (§4.3). We plan to open-source our code (Reproducibility statement).
- **Extension: pruning the vision encoder and the language model together.** Most prior depth pruning of vision–language models removes only language-model layers (Ma et al., 2025). When the selection can take layers from both, the split of the budget decides which capabilities survive repair. Removing only language-model layers loses most of the grounding. Our probe sets the split from the measurement and keeps the best grounding at both sizes (§5).

2 Related work

Pruning criteria and recovery. Pruning removes weights (Frantar & Alistarh, 2023; Sun et al., 2024), structures (Michel et al., 2019; Ma et al., 2023; Ashkboos et al., 2024; Xia et al., 2024; Muralidharan et al., 2024), sub-blocks (Zhong et al., 2025) or whole blocks (Sajjad et al., 2023; Fan et al., 2020; Siddiqui et al., 2024), and merging or replacing blocks changes the architecture instead (Yang et al., 2024b; Chen et al., 2025c; Appendix O). The criteria of Table 1 differ in their statistic, but all measure damage to the unrepaired model. This is a weak guide. A model can look intact while specific capabilities are gone (Jaiswal et al., 2024), and across tens of thousands of retrained vision networks the loss before retraining barely predicts accuracy after it (Feng et al., 2026). Lu

Criterion	Predicts	Training	Score
ShortGPT (Men et al., 2025)	damage before repair	none	$1 - \cos\big(h_{u-1}, h_u\big)$
Angular (Gromov et al., 2025)	damage before repair	none	$\arccos\langle \hat{h}_{u-1}, \hat{h}_{u+T-1} \rangle$
SLEB (Song et al., 2024)	damage before repair	none	$\mathrm{PPL}(f_{-S})$, re-scored
Shortened LLaMA (Kim et al., 2024)	damage before repair	none	Taylor saliency or PPL
FinerCut (Zhang et al., 2024)	damage before repair	none	$\mathrm{JS}\big(p_f \,\|\, p_{f_{-S}}\big)$, re-scored
Reassessing (Lu et al., 2026)	—	none	none: the last T layers
DarwinLM (Tang et al., 2026)	recovery	per candidate	KL after a short fine-tune
Ours (Eq. 2)	recovery	none	d_k: JS with demonstrations

Table 1: **What each pruning criterion predicts.** *Predicts*: what the score stands in for. *Training*: what is trained to choose the layers. *Score*: T is the number of layers to remove, h_u the hidden state after layer u and $\hat{h}_u$ its unit-norm version. Damage scores are computed without demonstrations, so they cannot order candidates by what repair restores (Observation 1 in Appendix A; §4.2). DarwinLM trains every candidate. Our probe predicts recovery from in-context learning, without training.

et al. (2026) skip the measurement instead: after fine-tuning, removing the deepest layers beats elaborate criteria. Their rule fixes the position, whereas ours measures it, since the safe position moves with the backbone (§4.2). Two works estimate recovery by optimising each candidate. PRACTISE (Wang & Wu, 2024) fits adaptors around a dropped CNN block and calls the remaining error its *recoverability*, and DarwinLM (Tang et al., 2026) briefly fine-tunes each width-pruned candidate. We ask their question with forward passes, and a depth port of DarwinLM is a baseline (Appendix G.1). All these criteria assume one stack of identical blocks, which §5 drops.

In-context learning as implicit optimisation. In-context learning (Brown et al., 2020) has been interpreted as implicit Bayesian inference (Xie et al., 2022) and as implicit gradient descent, which is exact for linear attention (von Oswald et al., 2023) and argued for softmax transformers and pretrained LMs (Akyürek et al., 2023; Dai et al., 2023). Parts of the mechanism have been located, from induction heads to task and function vectors (Olsson et al., 2022; Hendel et al., 2023; Todd et al., 2024). How far the gradient-descent view holds is debated (Shen et al., 2024; Deutch et al., 2024), as is whether models learn new rules in context or recall old ones (Min et al., 2022; Wei et al., 2023b;a; Wu et al., 2024). Our probe's rules are new or counterfactual, so they cannot be recalled, only learned in context.

3 Method

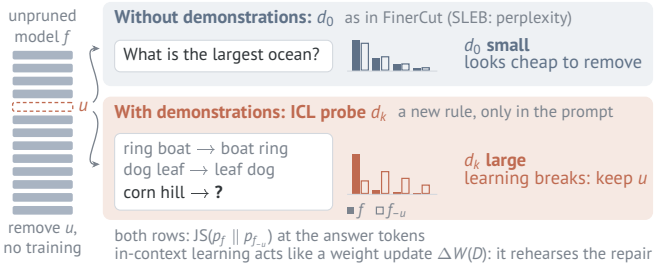
3.1 ICL for depth-pruning layer selection

Preliminary. A pruned model is usually repaired by fine-tuning, i.e., gradient descent on its remaining weights (Kim et al., 2024; Gromov et al., 2025). A line of work interprets in-context learning (ICL) as implicit optimisation (von Oswald et al., 2023; Akyürek et al., 2023; Dai et al., 2023). Let $f(x \mid D; W)$ denote the output of a model with weights W on an input x whose prompt contains demonstrations D. The view states that

$$f(x \mid D; W) \;=\; f\big(x \mid \varnothing; W + \Delta W(D)\big), \tag{1}$$

where $\varnothing$ denotes a prompt without demonstrations and $\Delta W(D)$ is the update induced by D, with $\Delta W(\varnothing) = 0$. For linear attention it holds exactly, since one head can implement one step of gradient descent on the in-context least-squares loss (von Oswald et al., 2023; Proposition 1). For large language models it remains a debated hypothesis (Dai et al., 2023; Shen et al., 2024; Deutch et al., 2024). Section 4.2 tests its prediction that the signal needs demonstrations (Appendix A).

Our method. Let f_{-S} denote the original model f with the layer set S removed, before any repair. Eq. 1 suggests viewing in-context learning as a rehearsal of repair. If the layers that turn demonstrations into $\Delta W(D)$ are also those that a few gradient steps rely on, a pruned model that still learns from its prompt should be easy to repair, and a removal that breaks this learning should



Algorithm 1 ICL-probe layer selection

Input: model f, budget T, probe items
1: $c(u) \leftarrow$ present damage of every u
default: $\mathrm{BI}(u) = 1 - \cos(h_{u-1}, h_u)$
2: $\mathcal{P} \leftarrow$ the $2T$ layers of smallest c
3: $\ell(u) \leftarrow d_k(\{u\})$ for $u \in \mathcal{P}$ (Eq. 2)
4: $S^\star \leftarrow$ the T layers of $\mathcal{P}$ of smallest ℓ
5: remove $S^\star$; repair once; **return**

Two towers (§5): $\mathcal{P}$ takes T layers from each tower, and lines 1–4 run once per removed layer, taking one layer each time, with no repair in between. Settings: Appendix B.

Figure 2: **Our method.** *Left*: a layer u is removed from the model f, and f and f_{-u} are compared by the JS divergence of their next-token distributions at the answer. Without demonstrations (d_0, FinerCut's statistic) it can look cheap. With demonstrations of a new rule, the ICL probe d_k (Eq. 2) asks whether the rest of the model can still learn it. *Right*: our implementation (§3.2).

be hard to undo. Existing criteria differ in their statistic, from the perplexity used by SLEB (Song et al., 2024) and the output divergence of FinerCut (Zhang et al., 2024) to the hidden-state similarity of ShortGPT (Men et al., 2025) and angular distance (Gromov et al., 2025). All of them, however, measure it on prompts without demonstrations, such as plain text or questions asked without examples (Appendix C). Their scores therefore depend only on $f_{-S}(\cdot \mid \varnothing; W)$, where the implicit update vanishes. Two layer sets whose removal changes the model identically on such prompts receive the same score, however differently the two pruned models learn in context (Observation 1).

We instead measure the Jensen–Shannon (JS) divergence between the two models with demonstrations in the prompt. Let D_k be a prompt with k demonstrations of a task followed by a query, and y its correct answer. We feed both models the same prompt and answer prefix, and write p_f and $p_{f_{-S}}$ for their next-token distributions at each token of y. We define

$$d_k(S) \;=\; \mathbb{E}\,\mathrm{JS}\big(p_f(\cdot \mid D_k) \,\big\|\, p_{f_{-S}}(\cdot \mid D_k)\big), \tag{2}$$

where the expectation is over answer tokens and probe items. With $k = 0$ and ordinary instructions, d_0 is the output divergence of FinerCut (Zhang et al., 2024) computed on our prompts, and it ranks the layers for our FinerCut baseline (§4.1). It captures only damage to what the model does without demonstrations. With $k > 0$ we use tasks the unpruned model cannot solve without the demonstrations (Table 5), so its answer must come from learning in context. Then d_k measures how much a removal damages that learning (Corollary 1). We call d_k the *ICL probe*. A small $d_k(S)$ means that f_{-S} still learns the task from its demonstrations, so S is a good candidate for removal.

3.2 Implementation

Candidates, then probe. A removal should keep what the model already does, and it should leave the remaining layers able to learn in context. Present-damage criteria such as block influence measure the first, and our probe measures the second. We therefore combine them in two steps (Algorithm 1). To remove T layers, a present-damage score c first proposes the $2T$ layers with the smallest c. Each candidate u is then scored by its probe damage $\ell(u) = d_k(\{u\})$ on the unpruned model, and the T with the smallest damage are removed together and repaired once:

$$\mathcal{P} = \mathcal{L}_{2T}(c), \qquad S^\star = \mathcal{L}_T\big(\ell \,\big|\, \mathcal{P}\big), \tag{3}$$

where $\mathcal{L}_m(g \mid A)$ denotes the m elements of A with the smallest g (all layers when A is omitted), so $S^\star$ minimises $\sum_{u \in S} \ell(u)$ over the T-subsets of $\mathcal{P}$ (Appendix E.1 probes whole sets instead). The pool width $2T$ is the method's only hyper-parameter. It only has to leave the probe a choice, so we set it once, to twice the budget, and did not tune it. Our default c is block influence (Men et al., 2025), computed as in its paper (Appendix C). Ranking by this default alone is ShortGPT, which is thus our ablation (Table 12). With candidates from angular distance or d_0, or with no candidate step, results stay within a few points of this default (Appendix F).

Probe tasks and cost. For language models we use seven task families with new or counterfactual rules, such as swapping a word pair (Figure 2). A family is admitted only if the unpruned model is near chance on it without demonstrations and solves it with them. We draw 320 items with one or

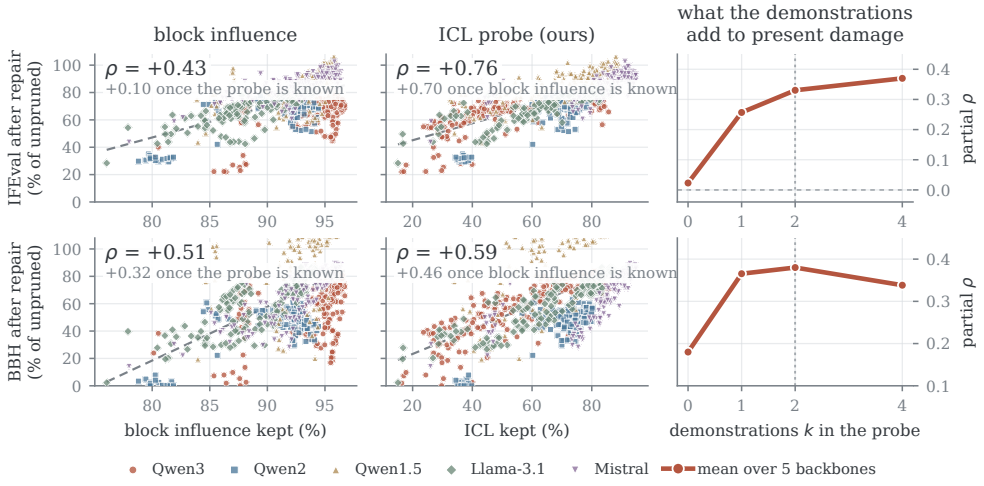


Figure 3: **Learnability predicts recovery better than present damage.** All 797 repaired sets at 25%. *Left, middle*: the mean BI of the removed layers and their summed probe score S (Eq. 3), before repair, as the share kept ($100(1 - \text{mean BI})$, $100\,e^{-S}$). *Right*: once BI is known, the probe's correlation with recovery rises when demonstrations are added ($k > 0$; contiguous and random sets, five-backbone mean; dotted: our k=2). More in Appendices E.4 and E.6.

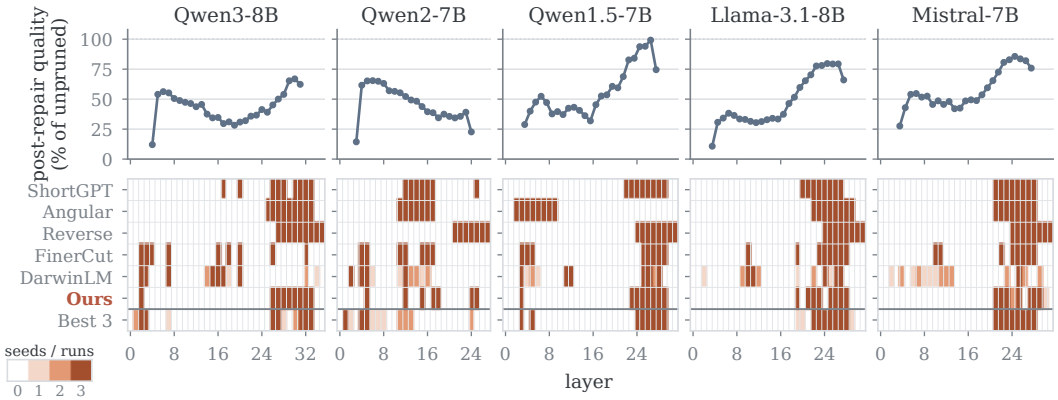


Figure 4: **Ours removes layers that repair well.** *Top*: we remove and repair every contiguous block of 25% of the layers, one point per block at its middle layer, scored by the post-repair mean of IFEval, GSM8K and BBH. **The best location differs by backbone.** *Bottom*: layers each criterion removed (colour: seeds, of three). The last row shows the 3 best on that mean of over 1K runs repaired once at 25% (191–211 per backbone).

two demonstrations (Appendix D). Replacing them with six generative tasks of the public ICLEval benchmark (Chen et al., 2025b) gives nearly the same five-backbone mean (Table 15). For vision–language models we use TrueMICL (Chen et al., 2025a), whose tasks cannot be solved from the demonstrations' text alone (§5). Selection needs only forward passes on a few hundred prompts and adds about 3% to the wall-clock of a prune-and-repair run (Appendix B).

4 Experiments and Analyses

4.1 Experimental Setup

Backbones and baselines. We prune five instruction-tuned 7–8B models with 28–36 layers: Qwen1.5-7B-Chat (Qwen Team, 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen3-8B

(Yang et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen2-7B-Instruct (Yang et al., 2024a). We compare our method with five baselines. **ShortGPT**: block influence, one minus the per-token cosine between a block's input and output. **Angular distance** (Gromov et al., 2025): the cheapest contiguous block of length T. **Reverse order**: the T deepest layers, as Lu et al. (2026) recommend. **FinerCut** (Zhang et al., 2024): the T cheapest layers by d_0, with no candidate pool. **DarwinLM** (Tang et al., 2026): a search that briefly fine-tunes each candidate, ported to depth (Appendix G.1).

Prune-and-repair protocol. Each criterion selects the T layers to remove before any repair. The pruned model then gets one LoRA repair (Hu et al., 2022) of $200T$ steps at batch 64 on Alpaca-cleaned (Taori et al., 2023) mixed with a Tülu-3 subset (Lambert et al., 2025) (Appendix B). At 25% removal this is 90K–115K examples, about as many as the repairs of Lu et al. (2026) and Kim et al. (2024). Every score is taken after this repair. All baselines run under this protocol. Wherever we could compare, it is at least as strong as the published recipe: for FinerCut on every backbone and axis (Table 4), and for block influence and, except on GSM8K, deepest-first removal (Appendix C).

Evaluation. Four axes: **MCQ** (twelve multiple-choice sets, mostly stored knowledge; per-set scores in Appendix N), **IFEval** (Zhou et al., 2023) for instruction following, and **GSM8K** (Cobbe et al., 2021) and **BBH** (Suzgun et al., 2023) for multi-step reasoning. More details on definitions and extractors are in Appendices B and B.3.

4.2 Which layers go decides recovery, and in-context learning predicts it

The removed set matters, and no fixed position is safe. On each backbone we removed and repaired well over a hundred different sets of 25% of the layers, including every contiguous block, random sets and sets near our own choice, all with the repair of §4.1 (the recovery map; Appendix B.4). At this fixed budget, post-repair quality varies far more across the sets than across the published criteria (Figure 1, left). Where the good sets lie also differs by backbone. Among contiguous blocks, post-repair quality peaks deep on four backbones and shallow on Qwen2-7B (Figure 4, top), and the same split appears under another repair schedule (Appendix E.5). The layers to remove therefore have to be chosen for each model.

Our selection is close to the best set we repaired. Among the contiguous and random sets, our selection is at or near the top on every backbone (Figure 1, left; Appendix E.2), and sets that resemble ours repair well (Appendix E.3). It also removes most of the best set's layers on four of the five backbones (Figure 4, bottom). A cheap measurement before repair thus finds near-best sets, and re-measuring after each repair does not narrow the gap (Appendix H).

The probe predicts recovery better than present-damage scores. Block influence, the signal behind ShortGPT (Men et al., 2025) and our default candidates, predicts recovery, but the summed probe score of the removed layers predicts it better on both axes, by a wide margin on IFEval (Figure 3; scope in Appendix E.7). The two scores are correlated, so we control for each in turn. Once the probe is known, block influence keeps little of its predictive power on IFEval and about two thirds on BBH. Once block influence is known, the probe retains most of its predictive power on both axes. FinerCut's statistic d_0 predicts recovery less well than either and keeps nothing once the probe is known (Appendix E).

The extra signal comes from the demonstrations. Eq. 1 predicts that the probe's advantage should shrink without demonstrations, because the implicit update then vanishes. Once block influence is known, the probe's correlation with IFEval after repair is 0.02 without demonstrations, 0.26 with one and 0.33 with two (Figure 3, right; Appendix E.4), and BBH shows the same rise. No present-damage score has this signal, since none uses demonstrations.

4.3 Comparing the criteria under the same repair

Five backbones, two repair objectives. Table 2 gives every criterion's selection the same repair budget, first as supervised fine-tuning and then as on-policy distillation (Agarwal et al., 2024), which trains the pruned model on its own generations to match the unpruned model. Under fine-tuning our

		Supervised Fine-Tuning (SFT)					On-Policy Distillation (OPD)				
Backbone	Criterion	MCQ_{12}	IFEval	GSM8K	BBH	Avg	MCQ_{12}	IFEval	GSM8K	BBH	Avg
Qwen3-8B	Unpruned	56.0	88.4	86.4	78.6	77.3	56.0	88.4	86.4	78.6	77.3
	ShortGPT	56.9 $_{\pm 0.7}$	62.4 $_{\pm 2.2}$	50.6 $_{\pm 2.0}$	51.7 $_{\pm 0.8}$	55.4 $_{\pm 1.2}$	46.8 $_{\pm 0.6}$	70.4 $_{\pm 2.1}$	62.4 $_{\pm 0.7}$	49.8 $_{\pm 2.1}$	57.3 $_{\pm 0.4}$
	Angular	57.8 $_{\pm 0.2}$	62.9 $_{\pm 1.2}$	44.3 $_{\pm 1.6}$	59.3 $_{\pm 1.3}$	56.1 $_{\pm 0.5}$	45.1 $_{\pm 0.1}$	72.0 $_{\pm 0.6}$	43.9 $_{\pm 1.4}$	62.1 $_{\pm 1.2}$	55.8 $_{\pm 0.4}$
	Reverse order	55.6 $_{\pm 0.4}$	51.5 $_{\pm 0.7}$	51.9 $_{\pm 0.4}$	54.0 $_{\pm 0.4}$	53.2 $_{\pm 0.3}$	46.1 $_{\pm 0.3}$	54.3 $_{\pm 1.3}$	64.6 $_{\pm 1.8}$	60.6 $_{\pm 2.0}$	56.4 $_{\pm 0.4}$
	FinerCut	57.4 $_{\pm 0.5}$	60.9 $_{\pm 1.7}$	52.7 $_{\pm 1.2}$	43.9 $_{\pm 0.7}$	53.7 $_{\pm 0.4}$	**51.9** $_{\pm 0.4}$	68.4 $_{\pm 1.7}$	63.8 $_{\pm 0.5}$	41.9 $_{\pm 1.8}$	56.5 $_{\pm 0.3}$
	DarwinLM	56.2 $_{\pm 0.8}$	55.4 $_{\pm 0.8}$	43.8 $_{\pm 2.3}$	35.3 $_{\pm 2.0}$	47.7 $_{\pm 1.1}$	50.6 $_{\pm 0.9}$	58.7 $_{\pm 1.5}$	56.5 $_{\pm 0.7}$	34.3 $_{\pm 1.0}$	50.0 $_{\pm 0.5}$
	Ours	**58.3** $_{\pm 0.1}$	**65.3** $_{\pm 2.4}$	**60.1** $_{\pm 0.6}$	**63.1** $_{\pm 0.2}$	**61.7** $_{\pm 0.7}$	45.9 $_{\pm 0.1}$	**75.6** $_{\pm 1.6}$	**68.6** $_{\pm 0.7}$	**66.3** $_{\pm 0.9}$	**64.1** $_{\pm 0.3}$
Qwen2-7B	Unpruned	61.5	62.2	76.8	60.9	65.3	61.5	62.2	76.8	60.9	65.3
	ShortGPT	51.3 $_{\pm 0.6}$	42.6 $_{\pm 1.2}$	25.0 $_{\pm 0.6}$	26.4 $_{\pm 1.1}$	36.3 $_{\pm 0.3}$	50.0 $_{\pm 0.2}$	43.7 $_{\pm 0.4}$	25.8 $_{\pm 1.0}$	22.2 $_{\pm 2.3}$	35.4 $_{\pm 0.8}$
	Angular	50.9 $_{\pm 0.6}$	43.6 $_{\pm 1.5}$	23.2 $_{\pm 1.3}$	26.5 $_{\pm 0.5}$	36.0 $_{\pm 0.6}$	51.0 $_{\pm 0.4}$	40.7 $_{\pm 0.9}$	23.2 $_{\pm 0.3}$	17.5 $_{\pm 2.2}$	33.1 $_{\pm 0.6}$
	Reverse order	45.3 $_{\pm 0.6}$	26.5 $_{\pm 0.4}$	2.9 $_{\pm 0.4}$	13.8 $_{\pm 1.1}$	22.1 $_{\pm 0.5}$	44.6 $_{\pm 0.5}$	22.6 $_{\pm 1.4}$	1.9 $_{\pm 0.2}$	1.7 $_{\pm 1.5}$	17.7 $_{\pm 0.3}$
	FinerCut	51.1 $_{\pm 0.1}$	46.4 $_{\pm 1.6}$	31.3 $_{\pm 0.5}$	27.9 $_{\pm 1.9}$	39.2 $_{\pm 0.7}$	51.3 $_{\pm 0.8}$	47.6 $_{\pm 2.2}$	33.9 $_{\pm 0.9}$	24.4 $_{\pm 0.6}$	39.3 $_{\pm 0.3}$
	DarwinLM	**54.1** $_{\pm 1.0}$	**48.2** $_{\pm 0.4}$	**36.6** $_{\pm 3.5}$	30.3 $_{\pm 2.6}$	**42.3** $_{\pm 1.2}$	**53.5** $_{\pm 1.2}$	**48.5** $_{\pm 1.3}$	**41.8** $_{\pm 3.2}$	**28.7** $_{\pm 1.5}$	**43.1** $_{\pm 1.5}$
	Ours	51.4 $_{\pm 0.4}$	47.3 $_{\pm 0.4}$	28.7 $_{\pm 1.2}$	**30.4** $_{\pm 0.6}$	39.4 $_{\pm 0.5}$	50.4 $_{\pm 0.8}$	46.5 $_{\pm 2.1}$	26.9 $_{\pm 0.3}$	26.7 $_{\pm 1.2}$	37.6 $_{\pm 0.5}$
Qwen1.5-7B	Unpruned	56.3	50.6	57.6	27.9	48.1	56.3	50.6	57.6	27.9	48.1
	ShortGPT	53.4 $_{\pm 0.3}$	47.8 $_{\pm 1.6}$	28.9 $_{\pm 0.6}$	39.2 $_{\pm 0.6}$	42.3 $_{\pm 0.4}$	51.0 $_{\pm 0.3}$	43.0 $_{\pm 1.6}$	19.2 $_{\pm 1.0}$	22.6 $_{\pm 1.1}$	34.0 $_{\pm 0.5}$
	Angular	46.6 $_{\pm 0.4}$	38.0 $_{\pm 0.1}$	19.4 $_{\pm 1.3}$	11.2 $_{\pm 1.4}$	28.8 $_{\pm 0.6}$	45.3 $_{\pm 0.4}$	28.8 $_{\pm 1.7}$	16.9 $_{\pm 1.7}$	8.6 $_{\pm 2.1}$	24.9 $_{\pm 0.8}$
	Reverse order	52.7 $_{\pm 1.0}$	38.0 $_{\pm 1.3}$	27.5 $_{\pm 0.7}$	29.3 $_{\pm 0.6}$	36.9 $_{\pm 0.7}$	50.4 $_{\pm 0.1}$	39.9 $_{\pm 2.0}$	26.4 $_{\pm 1.1}$	21.3 $_{\pm 0.3}$	34.5 $_{\pm 0.8}$
	FinerCut	53.2 $_{\pm 0.5}$	51.5 $_{\pm 0.5}$	**37.5** $_{\pm 0.7}$	36.8 $_{\pm 0.5}$	**44.8** $_{\pm 0.4}$	51.7 $_{\pm 0.4}$	45.4 $_{\pm 0.8}$	**39.5** $_{\pm 0.8}$	21.6 $_{\pm 1.2}$	**39.5** $_{\pm 0.3}$
	DarwinLM	52.6 $_{\pm 1.7}$	50.2 $_{\pm 2.4}$	29.2 $_{\pm 1.5}$	31.9 $_{\pm 2.4}$	41.0 $_{\pm 2.0}$	51.1 $_{\pm 0.6}$	44.5 $_{\pm 0.9}$	31.3 $_{\pm 1.3}$	23.7 $_{\pm 4.9}$	37.6 $_{\pm 1.9}$
	Ours	**54.1** $_{\pm 0.1}$	**51.6** $_{\pm 2.0}$	30.5 $_{\pm 0.8}$	**41.8** $_{\pm 1.1}$	44.5 $_{\pm 0.2}$	**51.8** $_{\pm 0.1}$	**46.0** $_{\pm 1.4}$	27.3 $_{\pm 0.5}$	**27.8** $_{\pm 0.9}$	38.2 $_{\pm 0.4}$
Llama-3.1-8B	Unpruned	64.8	81.8	76.0	72.1	73.7	64.8	81.8	76.0	72.1	73.7
	ShortGPT	59.9 $_{\pm 0.5}$	64.8 $_{\pm 1.3}$	51.6 $_{\pm 1.1}$	60.6 $_{\pm 0.3}$	59.2 $_{\pm 0.4}$	**58.5** $_{\pm 0.3}$	69.8 $_{\pm 0.7}$	59.6 $_{\pm 0.4}$	58.1 $_{\pm 0.1}$	61.5 $_{\pm 0.2}$
	Angular	**60.9** $_{\pm 0.2}$	**67.1** $_{\pm 0.8}$	**53.1** $_{\pm 1.1}$	**62.3** $_{\pm 0.3}$	**60.8** $_{\pm 0.3}$	58.1 $_{\pm 0.1}$	71.6 $_{\pm 0.7}$	**64.3** $_{\pm 1.4}$	**61.9** $_{\pm 0.1}$	**64.0** $_{\pm 0.2}$
	Reverse order	59.4 $_{\pm 0.1}$	49.0 $_{\pm 2.7}$	49.3 $_{\pm 0.3}$	54.6 $_{\pm 1.4}$	53.1 $_{\pm 0.8}$	54.0 $_{\pm 0.2}$	41.6 $_{\pm 0.9}$	50.8 $_{\pm 4.1}$	39.1 $_{\pm 0.9}$	46.4 $_{\pm 1.2}$
	FinerCut	59.1 $_{\pm 0.3}$	66.1 $_{\pm 2.1}$	50.6 $_{\pm 1.7}$	51.5 $_{\pm 0.5}$	56.8 $_{\pm 0.6}$	57.1 $_{\pm 0.2}$	71.9 $_{\pm 0.5}$	56.8 $_{\pm 0.6}$	49.4 $_{\pm 0.9}$	58.8 $_{\pm 0.2}$
	DarwinLM	56.9 $_{\pm 1.2}$	61.0 $_{\pm 2.7}$	37.9 $_{\pm 5.4}$	36.3 $_{\pm 2.6}$	48.0 $_{\pm 2.9}$	55.0 $_{\pm 1.6}$	62.0 $_{\pm 3.7}$	43.0 $_{\pm 5.7}$	32.9 $_{\pm 4.2}$	48.2 $_{\pm 3.7}$
	Ours	59.6 $_{\pm 0.4}$	66.7 $_{\pm 0.9}$	50.4 $_{\pm 1.4}$	60.9 $_{\pm 0.1}$	59.4 $_{\pm 0.5}$	57.9 $_{\pm 0.0}$	**73.7** $_{\pm 0.7}$	59.3 $_{\pm 0.3}$	52.8 $_{\pm 0.3}$	60.9 $_{\pm 0.2}$
Mistral-7B	Unpruned	64.2	59.4	51.0	55.7	57.6	64.2	59.4	51.0	55.7	57.6
	ShortGPT	56.2 $_{\pm 0.7}$	**56.6** $_{\pm 0.9}$	35.6 $_{\pm 1.9}$	**49.5** $_{\pm 0.6}$	**49.5** $_{\pm 0.5}$	**58.6** $_{\pm 0.3}$	48.4 $_{\pm 0.9}$	18.3 $_{\pm 1.3}$	49.1 $_{\pm 0.8}$	43.6 $_{\pm 0.4}$
	Angular	56.2 $_{\pm 0.7}$	**56.6** $_{\pm 0.7}$	35.4 $_{\pm 2.0}$	49.4 $_{\pm 0.4}$	49.4 $_{\pm 0.5}$	58.3 $_{\pm 0.2}$	50.3 $_{\pm 1.5}$	19.8 $_{\pm 0.5}$	**49.5** $_{\pm 1.1}$	44.5 $_{\pm 0.2}$
	Reverse order	56.7 $_{\pm 0.4}$	47.0 $_{\pm 1.4}$	33.9 $_{\pm 1.2}$	44.6 $_{\pm 1.3}$	45.5 $_{\pm 0.4}$	57.8 $_{\pm 0.0}$	38.1 $_{\pm 1.1}$	**26.0** $_{\pm 1.1}$	43.9 $_{\pm 0.7}$	41.5 $_{\pm 0.4}$
	FinerCut	54.9 $_{\pm 0.5}$	55.6 $_{\pm 1.9}$	32.5 $_{\pm 2.0}$	41.1 $_{\pm 0.5}$	46.0 $_{\pm 0.5}$	57.5 $_{\pm 0.0}$	48.2 $_{\pm 2.3}$	25.2 $_{\pm 1.1}$	39.7 $_{\pm 1.3}$	42.7 $_{\pm 0.2}$
	DarwinLM	52.0 $_{\pm 0.2}$	52.1 $_{\pm 0.8}$	29.6 $_{\pm 2.0}$	28.6 $_{\pm 1.9}$	40.6 $_{\pm 0.5}$	54.4 $_{\pm 1.4}$	44.8 $_{\pm 1.6}$	22.1 $_{\pm 2.6}$	27.4 $_{\pm 2.2}$	37.2 $_{\pm 0.5}$
	Ours	**56.9** $_{\pm 0.5}$	54.2 $_{\pm 1.0}$	**36.8** $_{\pm 2.6}$	47.9 $_{\pm 0.2}$	48.9 $_{\pm 0.4}$	58.1 $_{\pm 0.1}$	**50.8** $_{\pm 3.6}$	23.6 $_{\pm 1.4}$	48.2 $_{\pm 0.5}$	**45.2** $_{\pm 0.6}$
Mean of five	Unpruned	60.6	68.5	69.6	59.0	64.4	60.6	68.5	69.6	59.0	64.4
	ShortGPT	55.5 $_{\pm 0.3}$	54.8 $_{\pm 0.2}$	38.3 $_{\pm 0.1}$	45.5 $_{\pm 0.5}$	48.5 $_{\pm 0.1}$	53.0 $_{\pm 0.2}$	55.1 $_{\pm 0.3}$	37.1 $_{\pm 0.3}$	40.4 $_{\pm 0.9}$	46.4 $_{\pm 0.2}$
	Angular	54.5 $_{\pm 0.2}$	53.6 $_{\pm 0.5}$	35.1 $_{\pm 0.2}$	41.7 $_{\pm 0.6}$	46.2 $_{\pm 0.1}$	51.6 $_{\pm 0.1}$	52.7 $_{\pm 0.1}$	33.6 $_{\pm 0.6}$	39.9 $_{\pm 0.8}$	44.4 $_{\pm 0.2}$
	Reverse order	53.9 $_{\pm 0.3}$	42.4 $_{\pm 0.7}$	33.1 $_{\pm 0.1}$	39.3 $_{\pm 0.6}$	42.2 $_{\pm 0.3}$	50.6 $_{\pm 0.1}$	39.3 $_{\pm 0.4}$	33.9 $_{\pm 0.8}$	33.3 $_{\pm 0.8}$	39.3 $_{\pm 0.4}$
	FinerCut	55.1 $_{\pm 0.1}$	56.1 $_{\pm 0.9}$	40.9 $_{\pm 0.6}$	40.2 $_{\pm 0.5}$	48.1 $_{\pm 0.3}$	**53.9** $_{\pm 0.2}$	56.3 $_{\pm 1.2}$	**43.8** $_{\pm 0.3}$	35.4 $_{\pm 0.9}$	47.4 $_{\pm 0.1}$
	DarwinLM	54.4 $_{\pm 0.3}$	53.4 $_{\pm 0.9}$	35.4 $_{\pm 2.3}$	32.5 $_{\pm 1.6}$	43.9 $_{\pm 1.2}$	52.9 $_{\pm 0.6}$	51.7 $_{\pm 0.5}$	38.9 $_{\pm 2.3}$	29.4 $_{\pm 1.9}$	43.2 $_{\pm 1.3}$
	Ours	**56.1** $_{\pm 0.2}$	**57.0** $_{\pm 0.4}$	**41.3** $_{\pm 0.9}$	**48.8** $_{\pm 0.2}$	**50.8** $_{\pm 0.3}$	52.8 $_{\pm 0.1}$	**58.5** $_{\pm 0.1}$	41.1 $_{\pm 0.3}$	**44.4** $_{\pm 0.4}$	**49.2** $_{\pm 0.1}$

Table 2: **Main results at 25% removal: our method has the best average in both halves.** Accuracy (%), three seeds. Every criterion gets the same LoRA repair, so rows differ only in selection. **Left**: supervised fine-tuning; **right**: on-policy distillation from the unpruned model (Appendix J). *Unpruned*: the original model. Avg: mean of the columns. Last block: five-backbone mean. Subscripts: seed standard deviation. Bold: column best. Repair data alone: Table 22.

method leads the five-backbone mean on every axis, and under distillation it leads the average, IFEval and BBH. In both halves its average is about 2 points above the strongest baseline. Beyond 25% removal, distillation repairs worse than fine-tuning on every axis of the five-backbone mean (Figure 9), so we compare under it only at 25% (Appendix J).

Where the gains come from. Every criterion gets the same repair, so a margin can only come from the selected layers, and our method wins where its set departs from the baselines'. On the three Qwen backbones the sets differ, and under fine-tuning it is the best training-free criterion on every axis except GSM8K on two of them. On Qwen3-8B, for example, it removes layer 2 with the last layers, a set no baseline selects (Figure 4). On Llama-3.1-8B and Mistral-7B our set shares most layers with block influence and angular distance, and the scores stay within a few points of the best baseline. Briefly fine-tuning every candidate does not move DarwinLM away from damage. Its

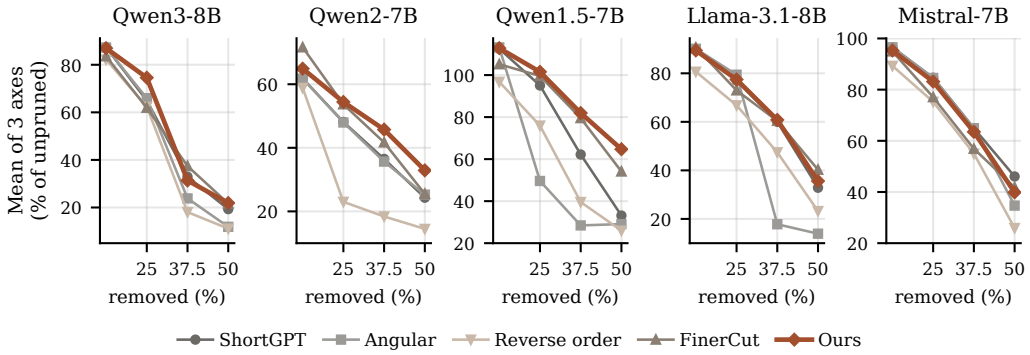


Figure 5: **From 25 to 50% removal, our method keeps the most on average.** Share of the unpruned model kept after repair, against the fraction of layers removed, three seeds. Each curve is the mean of IFEval, GSM8K and BBH. Multiple choice is left out because it separates the criteria far less than the other axes. Per axis: Table 16.

layers stay close to FinerCut's (Appendix G.1), and under fine-tuning it trails our method on every axis of four backbones.

Across removal rates. At 25% removal, our method is ahead of FinerCut on every backbone on the three-axis mean of Figure 5. At higher rates it stays ahead on average but not on every backbone, and it beats the other baselines on most backbones. The two lead on different axes, ours on BBH and FinerCut on GSM8K (Appendix G, Table 16). In the worst case over backbones, ours has the best BBH at 25 and 50%, while on IFEval and GSM8K no criterion is best at both higher rates.

What the probe adds, and what it costs. Without the probe, our method reduces to ShortGPT, so the ShortGPT rows show what the probe adds. On the five-backbone mean the probe improves every axis, the reasoning axes most, and Qwen3-8B gains most (Table 12). The gain comes from the probe itself. With no candidate pool at all, the probe alone beats every present-damage criterion on the mean and in the worst case over backbones (Table 14). The probe matters more as more layers go, up to half the network. Its reasoning gain peaks at 50% removal (Table 13), and the larger models of Appendix M gain more at 50% than at 25%. Whether the criteria agree is not known in advance, so the cost when the probe does not help matters. On the five-backbone mean it costs at most about a point on any axis and rate, and a few points on the worst backbone (Table 13).

Further analysis. BBH is scored with the three chain-of-thought exemplars of prior work, the only exemplars in our evaluation. Since our criterion is itself in-context, we also score BBH zero-shot, and the overall trend is the same (Appendix I.2). Appendix M repeats these comparisons on language models from 1.8B to 32B. At double the repair budget our lead on the reasoning axes widens, and only IFEval goes to FinerCut (Appendix I).

5 Extending to Multimodal LLMs

Most existing depth pruning of multimodal LLMs removes only language-tower blocks (Ma et al., 2025), and cross-tower budgets have been set for the weights (Sung et al., 2024) and channels (Shi et al., 2023) of vision–language models but not for their blocks. Here **the candidate pool spans both towers**, so layers of the two towers compete for one budget under one criterion, and the main choice is how many layers each tower gives up.

Setup. We follow §4.1 on InternVL3.5 at 2B and 8B, repairing both towers with LoRA on MAmmoTH-VL data (Guo et al., 2025; Appendix L). Reverse (ViT) and Reverse (LM) fix the tower, and the rest rank both towers together. Since a shared pool is dominated by the cheaper tower, we build one pool per tower and our method and FinerCut re-measure the score after each

		Pruned		TTFT	SFT					OPD				
Backbone	Criterion	ViT	LM	(ms)	MCQ_8	$Text_4$	$Long_2$	$Ground_2$	Avg	MCQ_8	$Text_4$	$Long_2$	$Ground_2$	Avg
InternVL3.5-2B	Unpruned	0	0	114	64.3	75.6	68.0	24.7	58.1	64.3	75.6	68.0	24.7	58.1
	ShortGPT	0	8	99 (−15)	49.1 $_{\pm 0.4}$	57.2 $_{\pm 0.6}$	33.3 $_{\pm 1.5}$	8.5 $_{\pm 1.3}$	37.0 $_{\pm 0.1}$	48.3 $_{\pm 0.3}$	57.4 $_{\pm 0.4}$	37.6 $_{\pm 0.3}$	4.5 $_{\pm 1.5}$	36.9 $_{\pm 0.3}$
	Angular	1–2	6–7	99 (−15)	54.6 $_{\pm 0.1}$	60.6 $_{\pm 1.1}$	32.7 $_{\pm 1.9}$	5.3 $_{\pm 2.2}$	38.3 $_{\pm 1.0}$	53.9 $_{\pm 0.3}$	61.2 $_{\pm 1.6}$	40.3 $_{\pm 2.8}$	3.4 $_{\pm 4.6}$	39.7 $_{\pm 2.2}$
	Reverse (ViT)	8	0	98 (−16)	57.6 $_{\pm 0.2}$	**66.1** $_{\pm 0.1}$	39.9 $_{\pm 1.2}$	21.1 $_{\pm 1.7}$	46.2 $_{\pm 0.5}$	57.7 $_{\pm 0.4}$	67.4 $_{\pm 0.1}$	45.6 $_{\pm 0.4}$	20.8 $_{\pm 0.3}$	**47.9** $_{\pm 0.2}$
	Reverse (LM)	0	8	99 (−15)	57.1 $_{\pm 0.6}$	46.2 $_{\pm 0.1}$	25.3 $_{\pm 0.9}$	8.4 $_{\pm 0.7}$	34.3 $_{\pm 0.5}$	55.4 $_{\pm 0.4}$	45.8 $_{\pm 0.3}$	27.0 $_{\pm 0.8}$	1.1 $_{\pm 0.9}$	32.3 $_{\pm 0.6}$
	FinerCut	6	2	99 (−15)	55.4 $_{\pm 1.2}$	**66.1** $_{\pm 0.8}$	42.0 $_{\pm 1.8}$	19.4 $_{\pm 0.7}$	45.7 $_{\pm 0.1}$	55.9 $_{\pm 1.5}$	**67.9** $_{\pm 0.9}$	46.0 $_{\pm 0.1}$	19.7 $_{\pm 0.4}$	47.4 $_{\pm 0.7}$
	Ours	8	0	97 (−17)	**58.2** $_{\pm 0.5}$	62.1 $_{\pm 0.3}$	**44.4** $_{\pm 0.6}$	**23.2** $_{\pm 0.6}$	**47.0** $_{\pm 0.2}$	**58.7** $_{\pm 0.2}$	63.6 $_{\pm 0.3}$	**47.2** $_{\pm 1.0}$	**22.1** $_{\pm 0.2}$	**47.9** $_{\pm 0.3}$
InternVL3.5-8B	Unpruned	0	0	222	70.6	79.6	74.9	38.3	65.8	70.6	79.6	74.9	38.3	65.8
	ShortGPT	1	7	190 (−32)	56.4 $_{\pm 0.3}$	66.9 $_{\pm 0.5}$	47.4 $_{\pm 1.0}$	23.0 $_{\pm 0.9}$	48.4 $_{\pm 0.2}$	55.6 $_{\pm 1.2}$	66.2 $_{\pm 0.1}$	51.9 $_{\pm 1.7}$	24.5 $_{\pm 2.1}$	49.6 $_{\pm 1.1}$
	Angular	5	3	200 (−22)	64.7 $_{\pm 3.5}$	73.7 $_{\pm 2.3}$	49.1 $_{\pm 2.1}$	25.2 $_{\pm 4.0}$	53.2 $_{\pm 2.5}$	65.1 $_{\pm 3.4}$	75.3 $_{\pm 2.2}$	55.6 $_{\pm 2.1}$	29.5 $_{\pm 4.0}$	56.4 $_{\pm 2.8}$
	Reverse (ViT)	8	0	207 (−15)	67.0 $_{\pm 0.5}$	75.1 $_{\pm 0.1}$	49.3 $_{\pm 1.4}$	32.0 $_{\pm 0.7}$	55.8 $_{\pm 0.1}$	66.9 $_{\pm 0.1}$	75.3 $_{\pm 0.4}$	56.0 $_{\pm 1.0}$	35.6 $_{\pm 1.0}$	58.4 $_{\pm 0.1}$
	Reverse (LM)	0	8	187 (−35)	68.1 $_{\pm 0.4}$	60.4 $_{\pm 0.7}$	39.4 $_{\pm 2.1}$	14.2 $_{\pm 2.9}$	45.5 $_{\pm 0.8}$	**68.8** $_{\pm 0.1}$	59.5 $_{\pm 0.4}$	47.1 $_{\pm 0.9}$	4.0 $_{\pm 1.4}$	44.9 $_{\pm 0.4}$
	FinerCut	6–7	1–2	203 (−19)	**68.2** $_{\pm 0.4}$	**75.6** $_{\pm 1.0}$	51.5 $_{\pm 1.0}$	30.7 $_{\pm 3.1}$	**56.5** $_{\pm 1.0}$	68.3 $_{\pm 0.2}$	**76.4** $_{\pm 0.9}$	54.9 $_{\pm 1.2}$	34.3 $_{\pm 2.0}$	58.5 $_{\pm 0.9}$
	Ours	7	1	204 (−18)	67.6 $_{\pm 0.6}$	74.0 $_{\pm 0.2}$	**51.6** $_{\pm 1.7}$	**32.7** $_{\pm 3.2}$	**56.5** $_{\pm 0.3}$	67.5 $_{\pm 0.1}$	75.0 $_{\pm 0.0}$	**56.3** $_{\pm 0.5}$	**36.9** $_{\pm 1.2}$	**58.9** $_{\pm 0.4}$

Table 3: **On two towers, our split keeps the best grounding.** Eight layers removed from a two-tower pool, one 768-step repair per criterion, three seeds. **Left**: fine-tuning; **right**: on-policy distillation on the same selections. *Pruned*: layers taken per tower (a range where seeds disagree). *Cost*: median time-to-first-token and its change. Column subscripts count benchmark sets (sixteen in all); $Ground_2$ is PointArena and Where2Place. Avg as in Table 2. Cell subscripts: seed standard deviation. Bold: best of six. Setup: Appendix L.1; repair-data control: Table 23.

removal. Accumulating beats a single pass over the same pool on every backbone (Table 17), as the second half of Observation 2 predicts. More details are in Appendix H.

Where the removals are spent decides which capability is lost. In Table 3 (left), criteria that take 6 or more of the 8 layers from the language tower lose far more grounding than those that leave it nearly intact. Reverse (LM), which takes all 8 layers there, is last on text reading on both backbones and last on grounding on 8B. Our method leads 3 of the 4 columns on 2B, and on 8B it ties FinerCut on the average, each winning two columns. Distillation (Table 3, right) leaves the 2B ordering unchanged, and on 8B ours wins grounding and keeps the same text-reading gap to FinerCut. Text reading is the only column where removing mostly vision layers is consistently costly, on both backbones and objectives. In the probe ablation, the ShortGPT row, block influence puts most of the budget in the language tower. The probe reverses the split, which adds 4–15 points on all 8 columns, most on 2B grounding. Both selections are stable across seeds, so the gap is due to the probe. The language tower holds most of the parameters, but for time to first token, removing 8 vision layers saves as much as removing 8 language layers at 2B and a little under half as much at 8B, because the vision tower runs on every image patch before the connector merges them.

Further analysis. With 12 and 16 layers removed, our margin over most baselines widens, and on each method's worst column we beat the other baselines at every budget and FinerCut in 4 of the 6 budget–size cells (Appendix L.2, Table 24). The cells we lose are mostly text reading, to FinerCut. On the larger InternVL3.5-14B and 38B, the probe again moves most removals to the vision tower and raises the average (Appendix M). On a second family, Qwen2-VL, the mechanism transfers but the ranking does not. The probe beats its ablation in 7 of 8 cells, but ours leads only one axis per size (Appendix M.2).

6 Conclusion and Limitations

Whether repair can undo a removal depends on what the remaining layers can still learn. We measure this with in-context learning, using forward passes only, and find that the probe helps predict post-repair quality better than present-damage scores alone. Combining it with block-influence candidates improves layer selection. At 25% removal our selection has the best five-model average under both repair objectives (SFT and OPD), and on vision–language models it chooses the two-tower split that best preserves grounding. Two limits follow from the design. The probe needs a backbone that already learns from its prompt: on Qwen1.5-1.8B, it adds nothing over ShortGPT (Appendix M). And our conclusions apply to the middle of the removal range, where the choice of layers decides what survives (§4.3). A natural next step is to test the probe on hybrid backbones that mix attention with state-space layers, and after other changes that need repair, such as quantisation.

REPRODUCIBILITY STATEMENT

An anonymised repository accompanies this submission for review: `https://anonymous.4open.science/r/ICL4DepthPruning-56F1`. It contains our selection method, the probe suite and the settings needed to reproduce the reported results; Appendices B and D give the protocol and the probe. We intend to release the full code after our institution's code-release review and to support reproduction of our results.

THE USE OF LARGE LANGUAGE MODELS

We used Claude (Anthropic) in the following ways. It helped write and polish the text. It searched for and summarised related work, which the authors then read. It helped state the propositions of Appendix A and write their proofs, which the authors checked. It wrote the generator of the synthetic probe items of Appendix D, and the authors reviewed the items before the admission gate chose the families. It also wrote the scripts that ran the experiments the authors designed, and it helped check the reported numbers and their descriptions against the experiment records. The research questions, the method and the hypotheses are the authors' own. The authors decided every interpretation in the paper, checked every claim and number, and take full responsibility for its content.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2306.13649.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2211.15661.

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*, 2024. URL https://arxiv.org/abs/2401.15024. arXiv:2401.15024.

Jean Berko. The Child's Learning of English Morphology. *Word*, 14(2–3):150–177, 1958. URL https://doi.org/10.1080/00437956.1958.11659661.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. Scene Text Visual Question Answering. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. URL https://arxiv.org/abs/1905.13648.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are We on the Right Way for Evaluating Large Vision-Language Models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2403.20330.

Shuo Chen, Jianzhe Liu, Zhen Han, Yan Xia, Daniel Cremers, Philip Torr, Volker Tresp, and Jindong Gu. True multimodal in-context learning needs attention to the visual context. In *Conference on Language Modeling (COLM)*, 2025a. arXiv:2507.15807.

Wentong Chen, Yankai Lin, ZhenHao Zhou, HongYun Huang, Yantao Jia, Zhao Cao, and Ji-Rong Wen. ICLEval: Evaluating In-Context Learning Ability of Large Language Models. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pp. 10398–10422, 2025b. URL https://aclanthology.org/2025.coling-main.693/.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining Redundant Layers to Compress Large Language Models. In *International Conference on Learning Representations*, 2025c. URL https://arxiv.org/abs/2403.19135. arXiv:2403.19135.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, Toronto, Canada, 2023. Association for Computational Linguistics. URL https://arxiv.org/abs/2212.10559.

Gilad Deutch, Nadav Magar, Tomer Natan, and Guy Dar. In-context Learning and Gradient Descent Revisited. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1017–1028, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://arxiv.org/abs/2311.07772.

Xuan Ding, Pengyu Tong, Ranjie Duan, Yunjian Zhang, Rui Sun, and Yao Zhu. Pruning as a Cooperative Game: Surrogate-Assisted Layer Contribution Estimation for Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2026.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing Transformer Depth on Demand with Structured Dropout. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1909.11556. arXiv:1909.11556.

Sicheng Feng, Keda Tao, and Huan Wang. Is Oracle Pruning the True Oracle? – A Sanity-Check of Neural Network Pruning with Retraining. *Transactions on Machine Learning Research (TMLR)*, 2026. URL https://arxiv.org/abs/2412.00143. arXiv:2412.00143.

Elias Frantar and Dan Alistarh. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2301.00774. arXiv:2301.00774.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The Language Model Evaluation Harness, 2026. URL https://zenodo.org/records/20122284.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The Unreasonable Ineffectiveness of the Deeper Layers. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. MAmmoTH-VL: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 13869–13920, 2025. URL https://arxiv.org/abs/2412.05237.

Roee Hendel, Mor Geva, and Amir Globerson. In-Context Learning Creates Task Vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, 2023. Association for Computational Linguistics. URL https://arxiv.org/abs/2310.15916.

Cristian Hinostroza, Rodrigo Toro Icarte, Christ Devia, Andres Carvallo De Ferari, Eugenio Herrera-Berg, Denis Parra, and Jorge F. Silva. Rethinking Layer Relevance in Large Language Models Beyond Cosine Similarity. In *International Conference on Learning Representations (ICLR)*, 2026.

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. URL https://arxiv.org/abs/2106.09685.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. Compressing LLMs: The Truth is Rarely Pure and Never Simple. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.01382.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened LLaMA: A Simple Depth Pruning for Large Language Models. In *ICLR Workshop on Mathematical and Empirical Understanding of Foundation Models (ME-FoMo)*, 2024. URL https://arxiv.org/abs/2402.02834.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023. URL https://arxiv.org/abs/2309.06180.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, et al. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2411.15124. arXiv:2411.15124.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2405.02246.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024. URL https://arxiv.org/abs/2305.07895.

Yao Lu, Yuqi Li, Wenbin Xie, Shanqing Yu, Qi Xuan, Zhaowei Zhu, and Shiping Wen. The Structural Scalpel: Automated Contiguous Layer Pruning for Large Language Models. *arXiv preprint arXiv:2510.23652*, 2025. URL https://arxiv.org/abs/2510.23652.

Yao Lu, Hao Cheng, Yujie Fang, Zeyu Wang, Jiaheng Wei, Dongwei Xu, Qi Xuan, and Zhaowei Zhu. Reassessing Layer Pruning in LLMs: New Insights and Methods. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2411.15558. arXiv:2411.15558.

Ji Ma, Wei Suo, Peng Wang, and Yanning Zhang. Short-LVLM: Compressing and accelerating large vision-language models by pruning redundant layers. In *ACM International Conference on Multimedia (ACM MM)*, 2025. URL https://arxiv.org/abs/2507.23362. arXiv:2507.23362.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the Structural Pruning of Large Language Models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.11627. arXiv:2305.11627.

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*, pp. 20192–20204, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-acl.1035/.

Paul Michel, Omer Levy, and Graham Neubig. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1905.10650. arXiv:1905.10650.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022. URL https://arxiv.org/abs/2202.12837. arXiv:2202.12837.

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. URL https://arxiv.org/abs/2407.14679.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context Learning and Induction Heads. *Transformer Circuits Thread*, 2022. URL https://arxiv.org/abs/2209.11895. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Qwen Team. Introducing Qwen1.5. https://qwenlm.github.io/blog/qwen1.5/, 2024. URL https://qwenlm.github.io/blog/qwen1.5/.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the Effect of Dropping Layers of Pre-trained Transformer Models. *Computer Speech & Language*, 77, 2023. URL https://arxiv.org/abs/2004.03844. arXiv:2004.03844.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained Transformers Learn In-Context by Gradient Descent? In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.08540.

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 (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31292–31311, 2023. URL https://arxiv.org/abs/2301.13741.

Shoaib Ahmed Siddiqui, Xin Dong, Greg Heinrich, Thomas Breuel, Jan Kautz, David Krueger, and Pavlo Molchanov. A Deeper Look at Depth Pruning of LLMs. *arXiv preprint arXiv:2407.16286*, 2024. URL https://arxiv.org/abs/2407.16286.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through Redundancy Verification and Elimination of Transformer Blocks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46136–46155. PMLR, 2024. URL https://proceedings.mlr.press/v235/song24f.html.

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*, 2024. URL https://arxiv.org/abs/2306.11695. arXiv:2306.11695.

Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. ECoFLaP: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.02998. arXiv:2310.02998.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023. URL https://arxiv.org/abs/2210.09261. arXiv:2210.09261.

Shengkun Tang, Oliver Sieberling, Eldar Kurtic, Zhiqiang Shen, and Dan Alistarh. DarwinLM: Evolutionary structured pruning of large language models. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2502.07780. arXiv:2502.07780.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023. URL https://github.com/tatsu-lab/stanford_alpaca.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function Vectors in Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.15213.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers Learn In-Context by Gradient Descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023. URL https://arxiv.org/abs/2212.07677.

Guo-Hua Wang and Jianxin Wu. Practical network acceleration with tiny sets: Hypothesis, theory, and algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9272–9285, 2024. doi: 10.1109/TPAMI.2024.3418999. URL https://arxiv.org/abs/2303.00972.

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. URL https://arxiv.org/abs/2409.12191.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv preprint arXiv:2508.18265*, 2025. URL https://arxiv.org/abs/2508.18265.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc Le. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 968–979, Singapore, 2023a. Association for Computational Linguistics. URL https://arxiv.org/abs/2305.08298.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023b. URL https://arxiv.org/abs/2303.03846.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1819–1862, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://arxiv.org/abs/2307.02477.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.06694.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080. arXiv:2111.02080.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*, 2024a.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yifei Yang, Zouying Cao, and Hai Zhao. LaCo: Large Language Model Pruning via Layer Collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6401–6417, Miami, Florida, USA, 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-emnlp.372/.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-naacl.51/.

Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. FinerCut: Finer-grained Interpretable Layer Pruning for Large Language Models. *arXiv preprint arXiv:2405.18218*, 2024. URL https://arxiv.org/abs/2405.18218. Compression Workshop at NeurIPS 2024.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. BlockPruner: Fine-grained Pruning for Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5065–5080, 2025. URL https://arxiv.org/abs/2406.10594. arXiv:2406.10594.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

CONTENTS OF THE APPENDIX

A Theory: why an in-context probe reads what repair needs . . . 16
B Full protocol . . . 18
C How strong are the baselines we reproduce? . . . 19
D The ICL probe: construction, admission gate, and example items . . . 20
E Further readings of the recovery map . . . 24
F Pool source and probe role . . . 28
G Rate curves and the tables at the other removal rates . . . 28
H When to score: once, or after each removal . . . 31
I Does the ordering survive more repair, or a different prompt? . . . 32
J Changing the repair objective: on-policy distillation . . . 33
K What the repair data does to an unpruned model . . . 34
L Vision–language experiments . . . 37
M Beyond the main backbones . . . 38
N Full per-set and per-cell tables . . . 41
O Prior selection signals, method by method . . . 42

A THEORY: WHY AN IN-CONTEXT PROBE READS WHAT REPAIR NEEDS

An attention layer can perform a gradient step on the examples in its prompt. This is exact for linear attention on in-context linear regression (Proposition 1) and a hypothesis in large language models, and the evidence that the probe predicts recovery is empirical (§4.2). The hypothesis explains why a score without demonstrations misses the in-context update, and where summing single-layer probes can fail (Proposition 2).

A.1 IN-CONTEXT LEARNING AS A GRADIENT STEP

In-context learning is learning from the prompt alone. The prompt shows a few worked examples of a rule the model has not seen, such as swapping the two words of a pair, and then a new case. A model that answers the new case correctly has learned the rule, although none of its weights has changed. The calculation below shows why this can look like a training step.

One gradient step. Take the simplest learning problem: learn a matrix W with $Wx \approx y$ from examples $(x_1, y_1), \ldots, (x_k, y_k)$. The squared error is $L(W) = \frac{1}{2}\sum_j \|Wx_j - y_j\|^2$, and its gradient is $\nabla L(W) = \sum_j (Wx_j - y_j)\, x_j^\top$. Start from $W = 0$, where the gradient is $-\sum_j y_j x_j^\top$. One gradient-descent step with step size η then gives $W = \eta \sum_j y_j x_j^\top$, and the updated model predicts, for a new input x,

$$\hat{y} \;=\; \eta \sum_{j=1}^{k} y_j \left(x_j^\top x\right). \tag{4}$$

Each example votes with its output y_j, weighted by how similar its input x_j is to the new input x.

One attention head. An attention head takes the same kind of weighted vote. Each earlier token offers a *key*, which says what it is about, and a *value*, which is what it passes on. The token being predicted offers a *query*. Without the softmax, the head returns $\sum_j v_j\,(k_j^\top q)$, the values weighted by how well their keys match the query. Now write each example as one token that holds both x_j and y_j. Let the head use x_j as the key, $\eta\, y_j$ as the value, and the new input x as the query. Its output is then exactly Eq. 4, the prediction of a model trained for one step on the prompt's examples.

Proposition 1 (one linear-attention head performs one gradient step; von Oswald et al., 2023)**.** *For in-context linear regression from $W = 0$, there are fixed key, query and value maps under which a single linear self-attention head outputs at the query exactly the prediction after one gradient-descent step on the in-context least-squares loss.*

The calculation above proves the proposition. The idea extends to deeper and trained transformers and, as implicit fine-tuning, to language models (von Oswald et al., 2023; Akyürek et al., 2023; Dai et al., 2023). How far it holds in large language models is debated (Shen et al., 2024; Deutch et al., 2024).

A.2 What this means for a pruned model

Eq. 1 states the same idea for a large model: reading the demonstrations D acts like adding an update $\Delta W(D)$ to the weights, which is zero without demonstrations. Repair also changes the weights, by real gradient steps on the repair data. Our hypothesis is that both kinds of learning rely on the same layers.

In practice, we give the unpruned model a prompt with a few examples of a made-up rule and a new case. We keep only tasks that the unpruned model fails without the examples and solves with them, so it can answer only by learning from the prompt. This check is the *admission gate* (Appendix D). Now remove a layer u. If the model still answers as before, the part of the network that learns from examples has survived, and by the hypothesis repair can still train the pruned model. The probe d_k of Eq. 2 measures this change. Two consequences follow.

Observation 1 (scores read without demonstrations cannot see the update)**.** *Assume Eq. 1. A selection score computed from a model's outputs on prompts without demonstrations sees only $f(\cdot \mid \varnothing; W)$, because with $D = \varnothing$ the update is zero. Two models that answer such prompts identically but learn differently from demonstrations get the same score.*

Every published depth-pruning signal has this form (Appendix O), including FinerCut's criterion d_0, which is our probe with no demonstrations.

Corollary 1 (what the admission gate ensures)**.** *Assume Eq. 1, and take a task family on which the unpruned model is at chance without demonstrations. Then between $k = 0$ and $k > 0$ demonstrations only $\Delta W(D)$ changes, and it is the only route to the answer.*

The gate holds on the unpruned model (Table 5) and on the criteria's sets (Table 6). Because the gate keeps the $k = 0$ accuracy at chance, the probe does not subtract a $k = 0$ term. The weights still read the prompt's words, so the probe also picks up some change in what the pruned model knows.

A.3 What the rule's two shortcuts miss

Algorithm 1 takes two shortcuts. It scores each layer alone and adds the scores, and it scores every layer once, on the unpruned model. Each shortcut has a cost.

Adding single-layer scores. Suppose two layers can each do the other's job. Removing either one alone costs nothing, but removing both breaks the model. Single-layer scores cannot see this, as the next statement makes precise.

Proposition 2 (the per-layer sum has a blind spot)**.** *Let $\ell(S)$ be the probe damage of removing the layer set S from a network whose in-context update is carried by a circuit C of $m \geq 2$ layers that keeps working as long as one of its layers survives. Then $\ell(\{u\}) = 0$ for every $u \in C$ while $\ell(C) > 0$, so no single-layer score can tell the layers of C from layers that do nothing. (Removing one $u \in C$ leaves $m - 1 \geq 1$ of its layers, so the update survives. Removing all of C removes it.)*

Probing the whole set would detect such a pair, but it also counts output damage that repair undoes. On the recovery map it predicts recovery no better than the sum (Appendix E.1).

Scoring once. Removing a layer changes how much the remaining layers matter, so one could re-score them after each removal instead.

Observation 2 (re-scoring after each removal changes the scores, for better or worse)**.** *Re-scoring the remaining layers of a pruned, unrepaired network measures each candidate in a model that will never be used without repair. On one stack of similar layers the scores tend to shift together, so re-scoring adds little information but still adds error, the same error that makes present-damage criteria misjudge recovery. On two towers, one tower's scores can rise faster than the other's, so re-scoring can show that the next removal belongs in the other tower, which a single pass cannot see.*

Appendix H measures both halves. Scoring once wins on one stack. On two towers the vision tower's costs rise about 3 times as fast as the language tower's, and re-scoring wins.

B FULL PROTOCOL

Settings of Algorithm 1. The candidate width is $2T$ (every layer once $2T \geq L$), set once and not tuned. The default candidate score, block influence, uses 64 Alpaca-cleaned prompts. The probe uses 320 items, at k=2 for rules induced from examples and k=1 for rules stated once (two towers: Appendix L.1). Repair is $200T$ LoRA steps in one run. Selection, repair, scoring and the recovery-map sets follow in B.1, B.2, B.3 and B.4.

B.1 SELECTION

At 25% removal every criterion except DarwinLM, which works in rounds, scores the original model, and the present-damage criteria share one calibration draw: 64 chat-formatted Alpaca-cleaned records for the two hidden-state criteria, and the same draw with gold continuations for FinerCut's d_0 references. Ours uses the probe sample of Appendix D. On the vision–language runs, all four criteria get the same 64 calibration images, drawn with the run seed from ST-VQA (Biten et al., 2019) as packaged in The Cauldron (Laurençon et al., 2024). Angular distance is read at the last token, as in its source, here the last position of the chat generation prompt.

What selection costs. Our comparisons hold the *repair* budget equal. At 25%, a complete prune-and-repair run costs about 3 per cent more with our selection than with the cheapest baseline (means of three seeds). The probe takes 1–2.5 minutes, against a repair of $200T$ steps. The comparison is uneven in two opposite ways. We spend more compute: the hidden-state baselines process the draw once, whereas FinerCut and ours ablate every candidate, with about $100\times$ more forward passes. We also use more input: only ours uses the probe suite, which a demonstration-free statistic has no use for. The suite is generated by rule, shares no item with any evaluation benchmark and is included in the anonymous repository (Reproducibility statement). Ours against FinerCut isolates its effect, since the two differ only in the candidate step and the demonstrations.

B.2 REPAIR AND SCHEDULES

Repair data. The language models are repaired on Alpaca-cleaned (the `yahma/alpaca-cleaned` release of the Stanford Alpaca data (Taori et al., 2023)) and an English subset of short examples from the Tülu-3 SFT mixture (Lambert et al., 2025), 154,363 examples in all. Examples are drawn without repetition and cut off at 256 tokens, and the loss is on response tokens only. The same data repairs every pruned model of a backbone and the unpruned model of Appendix K. The chat template is used throughout selection, repair and evaluation. The vision–language models are repaired on single-image instruction data (Appendix L.1).

Repair recipe. LoRA rank 8, α=16, dropout 0.05 on all 7 projection matrices; AdamW, learning rate 10^{-4}, no weight decay, 3% linear warm-up then cosine decay; batch 64; $200T$ optimiser steps for a budget of T layers, in one run on the final architecture, adapters merged at the end. At 25% removal (T = 7–9) the repair sees 89,600–115,200 examples, 12.9M–15.9M tokens, of which 7.3M–10.0M are the response tokens that carry the loss. For scale, Lu et al. (2026) fine-tune for two epochs of Alpaca-cleaned at batch 64, about 104K examples, and the LoRA repair of Kim et al. (2024) sees 100K (50K samples, two epochs). The iterative variant (Appendix H) spends the same total budget as 200 steps after each of T removals, with adapters merged and every signal recomputed on the current model after each round.

B.3 EVALUATION

Benchmarks and harness. Every benchmark is scored greedily, for pruned and unpruned models alike. No prompt holds demonstrations except BBH's three chain-of-thought exemplars (the zero-shot variant is in Appendix I.2). Language benchmarks run through `lm-evaluation-harness` v0.4.12 (Gao et al., 2026) on vLLM (Kwon et al., 2023) (bf16, one GPU per run; Qwen3 with think-

ing disabled). Vision–language benchmarks run through `lmms-eval` with each family's adapter and, where a judge is required, GPT-4o (2024-11-20).

The four language axes. MCQ is the mean of 12 multiple-choice sets (per-set scores in Appendix N). IFEval is instruction-level strict accuracy on all 541 items. GSM8K is zero-shot with chain of thought and numeric extraction (below). BBH runs on the first 200 items per subtask, with a format-tolerant answer extractor applied identically to every model.

Seeds. Every three-seed cell averages three independent seeds.

Which benchmarks count. A benchmark is kept only if the unpruned model scores sensibly under this protocol, with prompts and extractors identical for every criterion. CommonsenseQA fails this test. The unpruned Qwen3-8B scores at chance on it while any repaired model scores far above, so it measures the answer convention the repair data teaches rather than the capability.

Aggregating the axes. Avg is a plain mean over axes whose spreads differ, so an axis on which the criteria disagree widely counts for more. Ours is also first in both halves of Tables 2 and 3 under 4 other aggregates: the mean and the geometric mean of per-axis retention against the unpruned model, the mean rank over backbone–axis cells, and the worst-axis retention averaged over backbones. Against every baseline it wins most backbone–axis cells.

GSM8K scoring. The harness's default zero-shot GSM8K tasks stop generation at 256 tokens and let a LaTeX delimiter into the answer pattern. This penalises exactly the long markdown answers the unpruned models write. Half of the unpruned Qwen3-8B's answers stopped before the final number. Scoring at 1,024 tokens with a numeric-only pattern moves the unpruned models by 6 to 37 points and the 125 pruned cells by at most 1.2. The ordering among criteria is unaffected, but the default inflates every preservation ratio, whose denominator is the unpruned score. Every GSM8K number in the paper is scored this way.

B.4 The recovery-map collections

The recovery map (§4.2) removes and repairs each of its sets under the recipe of Appendix B.2, so its y-axis and Table 2 share one scale. Its 625 runs form five collections, each removing 25% of the layers: 125 contiguous blocks (25 per backbone); 160 random sets; 140 overlap-stratified sets; 125 one-layer swaps around a probe-chosen 25% set; and the 25% sets of the five criteria themselves (75 runs, three seeds on every backbone). Each set of the first four collections is one run with one seed. The same map scored with the probe of the whole set removed at once is in Appendix E.1, and its interleaved counterpart in Appendix E.5.

Other analyses draw on more runs, hence their larger totals. Figure 8 adds design variants of the pool and probe, and Figure 4 the remaining criterion runs. Figure 3 and Appendix E use every set repaired at 25% under this recipe. That is 1,646 runs over 797 distinct sets, the 543 sets of those four collections plus 254 that arose only as the choice of some criterion or design variant, with repeats averaged. Of those runs, 205 repaired between removals rather than once at the end. Dropping them leaves 782 sets and moves the probe's correlation by at most 0.005 on either axis.

C How strong are the baselines we reproduce?

Block influence and angular distance. A large-scale re-assessment of these criteria computes block influence from 10 BookCorpus samples truncated to 128 tokens (Lu et al., 2026), following Ma et al. (2023). We give every criterion 64 records on the language side and 64 images on the vision–language side (Appendix B), more than the 1–50 samples their own sweep covers. Their sweep also shows that the selection moves with the draw. All 5 of its sets differ, and the repaired averages span 0.40 to 0.43. We therefore fix the sample count once rather than search it, and give angular distance the same draw.

Our port does not weaken the criterion. On Qwen1.5-7B at 25% removal, with the same repair family (LoRA of rank 8 on Alpaca), the model pruned by the published block-influence recipe

answers knowledge questions at chance: MMLU 0.2456 and CMMLU 0.2536 against a chance level of 0.25, where reverse order reaches 0.5101 and 0.7171. Our port keeps nearly all of the unpruned multiple-choice score and is the strongest baseline on multiple choice there. The suites differ, so only this above-chance comparison carries over.

Where our baselines depart from their papers. Block influence is computed per token, as Men et al. (2025) define it. A whole-tensor variant found in some reimplementations selects an almost disjoint set of layers (Spearman 0.05 against the per-token ranking). FinerCut departs from its paper in 4 places. It removes whole layers rather than scoring attention and feed-forward blocks separately, a variant its own ablation reports. It ranks once rather than re-scoring greedily. It considers every layer, where the paper keeps to the last 60%. And it measures the divergence on our 64-record answer-position draw rather than on all positions of 10 Wikitext-2 samples. The next paragraph measures the effect of the calibration data.

Output-damage criteria under their published and aligned recipes. Table 4 repairs FinerCut and SLEB (Song et al., 2024) under our protocol, once with their published recipes (Wikitext-2) and once on our draw, one seed each. The published FinerCut row keeps its paper's data, positions, divergence and greedy search, on whole layers, with 64 samples and every layer a candidate. Under the published recipes the two repair to nearly the same scores, both far below the aligned FinerCut, so the difference comes from the calibration data rather than from the divergence. On instruction data SLEB improves but stays well behind the aligned FinerCut on every axis. Shortened LLaMA's perplexity criterion (Kim et al., 2024) is SLEB's score computed once rather than greedily, on BookCorpus, so the SLEB rows stand in for it.

Deepest-first under its own repair. The three Reverse rows of Table 4 repair the deepest-first set with our LoRA and with the partial fine-tune of Lu et al. (2026). At the published learning rate (10^{-5}), their partial fine-tune matches LoRA on IFEval and trails it on BBH on average, but scores higher on GSM8K on four of the five backbones. At our LoRA rate (10^{-4}) it lifts IFEval and overtakes ours on GSM8K on three backbones. At either rate it stays behind ours on instruction following and BBH on every backbone.

D THE ICL PROBE: CONSTRUCTION, ADMISSION GATE, AND EXAMPLE ITEMS

How the items are made. The probe is a suite of synthetic tasks from a small deterministic generator. No item text is model-generated. Each task *family* is a template with a rule and fixed word pools, and an item is a seeded draw from it. Rules are novel or counterfactual, so the answer cannot be retrieved from pretraining. Three families never state their rule, which must be induced from two input→output pairs (k=2). The other 4 state a counterfactual rule once, in a single worked example, and ask whether it persists into the query turn (k=1). In the three induction families the demonstrations and the query share no content word, which blocks lexical shortcuts.

Admission gate. A family is admitted only if, on all three intact backbones of the gate, it stays near chance without demonstrations and is solved with them. This enforces Corollary 1: whatever the probe measures must arrive through the demonstrations. Seven of 14 candidate families pass (Table 5). The others were answerable from the prior, or not learned from the demonstrations, by at least one backbone.

Suite and sample. The suite has 398 distinct items over 7 families. A run shuffles them with its seed and takes the first 320, so seeds differ in the draw but not in the families. The iterative variant keeps the first round's sample, and the original model's references, in every round. The draw matters on one backbone only. At 25% every seed removes the same layers on four backbones, and on Mistral-7B one seed takes a different deep layer from the other two. The seed also sets the repair's data order and adapter initialisation, so repair adds run-to-run variance of its own.

What the families test. Each family is anchored in a task the literature studies: duplication in the copy pattern that induction heads implement (Olsson et al., 2022), the two word-order families in

Backbone	Criterion	IFEval	GSM8K	BBH
Qwen3-8B	FinerCut, aligned (body)	61.0	53.9	43.7
	FinerCut, published recipe (Wikitext-2, all positions)	49.5	29.1	24.3
	SLEB, published recipe (Wikitext-2, NLL)	51.9	35.9	31.7
	SLEB, aligned data (chat answers, NLL)	60.7	31.1	37.0
	Reverse, LoRA repair (body)	50.8	51.6	54.2
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	58.4	54.4	46.1
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	47.8	40.3	46.9
	Ours	**67.5**	**60.8**	**62.9**
Qwen2-7B	FinerCut, aligned (body)	47.5	**30.9**	29.5
	FinerCut, published recipe (Wikitext-2, all positions)	39.3	21.4	24.6
	SLEB, published recipe (Wikitext-2, NLL)	39.3	21.6	24.7
	SLEB, aligned data (chat answers, NLL)	**48.4**	30.8	25.4
	Reverse, LoRA repair (body)	26.1	2.8	12.5
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	35.4	17.0	23.8
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	27.5	3.1	20.1
	Ours	47.2	29.9	**31.0**
Qwen1.5-7B	FinerCut, aligned (body)	**52.0**	37.7	37.2
	FinerCut, published recipe (Wikitext-2, all positions)	41.4	11.9	12.0
	SLEB, published recipe (Wikitext-2, NLL)	38.8	12.2	13.7
	SLEB, aligned data (chat answers, NLL)	43.9	18.6	30.8
	Reverse, LoRA repair (body)	37.3	26.8	29.1
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	45.8	**47.4**	26.0
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	38.4	37.5	24.4
	Ours	51.7	29.8	**41.9**
Llama-3.1-8B	FinerCut, aligned (body)	63.8	50.1	52.0
	FinerCut, published recipe (Wikitext-2, all positions)	57.9	35.9	34.6
	SLEB, published recipe (Wikitext-2, NLL)	52.9	29.2	29.5
	SLEB, aligned data (chat answers, NLL)	58.3	39.7	38.5
	Reverse, LoRA repair (body)	48.9	49.0	53.0
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	57.3	**62.9**	53.5
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	47.0	59.4	47.4
	Ours	**67.6**	50.8	**60.9**
Mistral-7B	FinerCut, aligned (body)	**53.8**	34.9	40.8
	FinerCut, published recipe (Wikitext-2, all positions)	50.5	29.0	33.8
	SLEB, published recipe (Wikitext-2, NLL)	48.8	30.6	34.2
	SLEB, aligned data (chat answers, NLL)	51.9	31.8	37.5
	Reverse, LoRA repair (body)	47.0	35.0	44.7
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	46.2	**40.5**	38.5
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	48.8	37.4	45.3
	Ours	53.1	39.7	**47.8**
Mean	FinerCut, aligned (body)	55.6	41.5	40.6
	FinerCut, published recipe (Wikitext-2, all positions)	47.7	25.5	25.9
	SLEB, published recipe (Wikitext-2, NLL)	46.4	25.9	26.8
	SLEB, aligned data (chat answers, NLL)	52.6	30.4	33.9
	Reverse, LoRA repair (body)	42.0	33.0	38.7
	Reverse, partial FT repair (last 3 layers + head), lr 10^{-4}	48.6	44.4	37.6
	Reverse, partial FT repair, lr 10^{-5} (Lu et al., 2026)	41.9	35.5	36.8
	Ours	57.4	42.2	48.9

Table 4: Output-damage criteria under their published recipes (Wikitext-2) and under our aligned calibration draw, with the body's aligned FinerCut and ours for reference. All are repaired under our protocol at 25%, one seed, accuracy (%). Bold: best of the eight rows within a backbone. The three Reverse rows share one removed set and differ only in its repair, with the same data and budget: LoRA (body), or full fine-tuning of the head and the last 3 layers (and the final norm, as in their released code) (Lu et al., 2026), at our LoRA rate (10^{-4}) or the published 10^{-5}.

		Qwen3-8B		Qwen1.5-7B		Mistral-7B	
Family	k	k=0	k	k=0	k	k=0	k
Duplication	2	0.00	0.78	0.00	1.00	0.08	0.98
Pair swap	2	0.00	0.67	0.02	0.92	0.05	0.94
Reorder with connective	2	0.00	0.89	0.00	0.94	0.00	1.00
Counterfactual operator	1	0.00	0.84	0.02	0.81	0.00	0.98
Novel lexicon	1	0.06	1.00	0.02	0.97	0.00	1.00
Format marker	1	0.00	1.00	0.00	1.00	0.00	0.89
Counterfactual world	1	0.00	1.00	0.06	0.89	0.00	0.80

Table 5: The admission gate for the seven families of the suite: accuracy without demonstrations (k=0) and at the family's working k (second column) on three intact backbones, 64 items per family. A family is admitted only if k=0 accuracy is ≤ 0.10 and accuracy at k is ≥ 0.6 on all three.

Backbone	Unpruned	ShortGPT	Angular	Reverse	FinerCut	**Ours**
Qwen3-8B	0.00	0.00	0.00	0.01	0.00	0.00
Qwen2-7B	0.00	0.00	0.00	0.00	0.00	0.01
Qwen1.5-7B	0.00	0.00	0.00	0.00	0.00	0.00
Llama-3.1-8B	0.00	0.00	0.00	0.01	0.01	0.00
Mistral-7B	0.00	0.00	0.00	0.01	0.00	0.00

Table 6: The gate on pruned candidates. Exact-match accuracy on the 320 probe items without demonstrations, measured on the unpruned model and on each criterion's 25% set before repair (one seed), greedy decoding. Every candidate stays at or below 0.01, so no pruned model answers the probe from the query alone.

the list transformations on which symbol tuning is evaluated (Wei et al., 2023a), the counterfactual operator in counterfactual tasks (Wu et al., 2024), the novel lexicon in the logic of the Wug test (Berko, 1958) and GPT-3's novel-word task (Brown et al., 2020), the format marker in the verifiable output constraints that IFEval scores (Zhou et al., 2023), and the counterfactual world in in-context information that overrides a semantic prior (Wei et al., 2023b). Table 7 states each family's rule.

Three probe corpora. **(i) The synthetic suite** makes the admission gate possible. We can discard a family that a backbone already solves at k=0. In the induction families we can also keep the vocabularies of demonstrations and query disjoint, which no public suite allows. **(ii) ICLEval** (Chen et al., 2025b) is a public suite built for the same question. When we swap it in, the rule's five-backbone mean barely moves (Table 15). Its `output_format` task (Format Cloning in the paper) takes half its items verbatim from GSM8K test questions, so we exclude it to keep the probe off an evaluation axis. **(iii) TrueMICL** (Chen et al., 2025a) is the vision–language probe (Figure 7). A synthetic image probe would be our own design twice over, in the rule and in the pixels. TrueMICL is built so that its demonstrations cannot be imitated from their text alone, which is what we need. Its Hugging Face card declares the MIT License.

The prompt the model sees. Items are plain chat turns with no instruction beyond the turns (Figure 6). The Qwen2 and Qwen1.5 templates add a one-line assistant prompt, and Llama-3.1's adds a date header.

What the two conditions look like. In Table 7, the unpruned model without demonstrations answers from its prior. It adds, names a likely colour, gives a dog's usual sound or explains the phrase. With the demonstrations it mostly follows the rule, though not always. Some misses are only a matter of form, such as a capital letter, which is why the selection statistic is a divergence at the gold tokens rather than exact match.

Scoring. Inside selection the statistic is *probe damage*, d_k of Eq. 2 at the family's working k: the same Jensen–Shannon divergence as d_0, in nats, between a reference model and the candidate prune, both teacher-forced on the gold continuation with the demonstrations in context, exactly where $\Delta W(D) \neq 0$. Appendix A.2 explains why no k=0 term is subtracted, and what this measurement

Pair swap (k=2, rule never stated)

```
<|im_start|>user
ring boat<|im_end|>
<|im_start|>assistant
boat ring<|im_end|>
<|im_start|>user
dog leaf<|im_end|>
<|im_start|>assistant
leaf dog<|im_end|>
<|im_start|>user
corn hill<|im_end|>
<|im_start|>assistant
<think>

</think>
```

gold: `hill corn`

Format marker (k=1, rule stated once)

```
<|im_start|>user
From now on, keep every answer to one
short line and always end it with
'..fin'. How many legs does a spider
have?<|im_end|>
<|im_start|>assistant
8 ..fin<|im_end|>
<|im_start|>user
How many hours are in a day?<|im_end|>
<|im_start|>assistant
<think>

</think>
```

gold: `24 ..fin`

Figure 6: The first pair-swap and format-marker items of the suite, as Qwen3-8B receives them: rendered by its chat template with thinking disabled, long lines wrapped to fit. The model continues the last assistant turn, and the probe scores the gold continuation with the demonstrations in context. *Left*: the rule is never stated and must be induced from the two demonstrations. *Right*: the rule is stated once, in the first turn, and must persist into the query turn.

Query	Gold	Without demonstrations	At the working k
Duplication: repeat the word (k=2)			
`wood`	`wood wood`	`It seems like you mentioned...`	*a reply in Chinese*
Pair swap: swap the two words (k=2)			
`corn hill`	`hill corn`	*a reply in Chinese*	`hill corn`
Reorder with connective: 'A and B' becomes 'B before A' (k=2)			
`rope and corn`	`corn before rope`	`"Rope and corn" is a phrase that...`	`corn before rope`
Counterfactual operator: the prompt redefines '+' (k=1)			
`Using that rule: 29 + 6 = ? Answer with the number only.`	`23`	`35`	`23`
Novel lexicon: a word defined once in the prompt (k=1)			
`What color is a 'lorp'? One word only.`	`yellow`	`Green`	`yellow`
Format marker: keep a stated format across turns (k=1)			
`How many hours are in a day?`	`24 ..fin`	`There are **24 hours in a day**.`	`24 ..fin`
Counterfactual world: a stated fact overrides the prior (k=1)			
`In this world, what sound does a dog make? One word only.`	`meow`	`Bark.`	`Meow.`

Table 7: The seven probe families, each with its rule in words. For the first suite item of each family: the query, the gold answer, and Qwen3-8B's greedy decodes without demonstrations and at the family's working k, made with the code of the admission gate (Table 5). Decodes are verbatim and truncated to fit. Replies in Chinese are described rather than reproduced. The decodes without demonstrations show the admission condition of Corollary 1.

does not isolate. The reference and the candidate depend on the schedule. In the single pass they are the original model and one candidate layer removed. In the iterative variant they are the original model and the current pruned-and-repaired model. In the accumulated rule of the vision–language runs they are the never-repaired original and the growing set. The vision–language probe draws 192 TrueMICL items, 48 from each of 4 tasks (operator induction, character classification, palindrome number, outlier detection). Each item has k=3 demonstration turns, and every turn carries its own image.

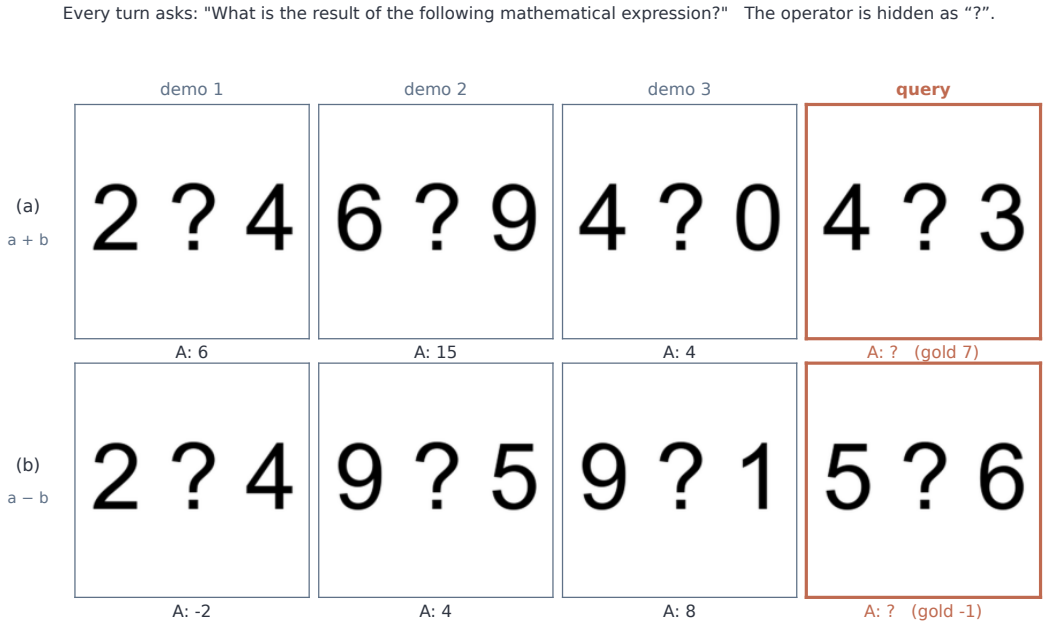


Figure 7: Two verbatim items of the vision–language probe: TrueMICL (Chen et al., 2025a), operator induction, k=3. Each image shows two operands with the operator masked, and every turn asks the same question. The two items open with the same image but answer it differently, so the rule can only come from the demonstrations and the operands only from the image. The operator under each label is our annotation. The probe draws 192 items evenly from 4 of TrueMICL's 7 tasks; Clock Math, CLEVR Count and Sudoku are left out.

E Further readings of the recovery map

The coefficients behind §4.2. Table 8 collects them, one row per control, and the probe is the strongest of the three signals in every row. The output divergence FinerCut optimises keeps nothing once the probe is held fixed, although it correlates with block influence at 0.54 and with the probe at 0.48. The probe keeps most of its correlation when block influence or the mean depth of the removed layers is held fixed. Both present-damage signals are weakest on contiguous blocks, the sets published depth-pruning methods remove, and the probe is strongest there. The pooled gap is therefore not an artefact of which sets we drew. On random sets the probe's lead on BBH is small.

E.1 Summing single-layer probes versus probing the whole set

Eq. 3 sums single-layer probes measured on the unpruned model. The alternative probes the whole set at once. The two agree on which region of the stack is expensive but disagree on sets that cut many layers near the output. Removing 8 of the 10 layers 21–30 of Mistral-7B collapses in-context learning, and the whole-set probe is 0.60 nats. The same 8 layers sum to 0.05, the cheapest of the 115 sets we trained on that backbone, and after repair the set recovers 86% of BBH, near the top of the collection. The whole-set probe mixes what repair cannot rebuild, the in-context machinery the layers held, with what repair undoes, the readout disruption that any deep cut causes.

		Block influence		d_0 (FinerCut)		ICL probe (ours)	
Sets and control	n	IF	BBH	IF	BBH	IF	BBH
All sets	550	.51	.52	.36	.29	**.76**	**.61**
other signal held	550	.13	.25	−.02	.00	**.67**	**.45**
mean depth held	550	.39	.35	.34	.27	**.69**	**.38**
Contiguous blocks only	125	.36	.46	.07	.08	**.79**	**.77**
Random sets only	160	.49	.48	.49	.43	**.71**	**.52**

Table 8: What predicts recovery, under five controls. Spearman ρ between a signal of the removed set and the set's post-repair share of the unpruned score, signed so that less damage is better (IF: IFEval). *All sets*: the 550 runs of the map's first four collections (Appendix B.4). *Other signal held*: partial correlation given the probe or, for the probe, given block influence (the two correlate at 0.57). *Mean depth held*: given the removed layers' mean relative depth. Bold: the best signal in the row. Figure 3 shows block influence and the probe on all 797 sets.

Over the recovery map (Table 9), the summed probe predicts recovery at least as well on both axes. On sets confined to the deeper half of the stack it predicts better on all five backbones (barely on Qwen2-7B and Mistral-7B), and there the whole-set probe reverses on Llama-3.1-8B and weakens sharply on Qwen3-8B. Our rule's sets rank first to fifth among their backbone's trained sets on the summed scale, but 12th to 94th on the whole-set scale. Among the criteria, they recover best on BBH on 3 of the five backbones and within 3 points of the best on the other two.

Two measurements of the unrepaired model mislead the same way. Re-scoring after each removal (Table 33) turns the whole-set probe into a selector. As the last layers are removed, the remaining deep layers look ever more expensive, so the rule moves to middle layers (layers 9 and 16 on Mistral-7B) and recovers worse on BBH. Few-shot accuracy misleads in the same way. On Mistral-7B our set answers 16% of the probe items with demonstrations, against 52% for ShortGPT's contiguous block, yet after the same repair the two land within two BBH points of each other (Table 2). In both cases the measurement is taken on the model before repair.

		IFEval, all sets		BBH, all sets		BBH, tail sets only	
Backbone	sets	whole set	summed	whole set	summed	sets	whole set / summed
Qwen3-8B	126	**0.78**	0.70	0.77	**0.84**	16	0.35 / **0.71**
Qwen2-7B	114	0.48	**0.61**	**0.45**	**0.45**	11	0.66 / **0.67**
Qwen1.5-7B	135	0.76	**0.80**	**0.77**	**0.77**	15	0.80 / **0.94**
Llama-3.1-8B	135	0.63	**0.86**	0.69	**0.94**	40	−0.32 / **0.75**
Mistral-7B	115	**0.79**	0.72	0.82	**0.93**	39	0.65 / **0.66**
Pooled	625	0.66	**0.75**	0.61	**0.63**		

Table 9: Two versions of the same probe on the recovery map of Figure 3. Spearman ρ with post-repair share, signed so that less damage is better, for the whole-set probe (the set removed at once) and the summed single-layer probe of Eq. 3. Sets: the 625 runs of the recovery map (Appendix B.4). Tail sets: every removed layer is in the deeper half of the stack. Bold: the higher of the two.

E.2 The ceiling in numbers

Table 10 gives the left panel of Figure 1 in numbers, for every backbone and with every set repaired by our recipe. The ceiling is sampled rather than exhaustive, so the percentile is optimistic. A set's score is the mean of IFEval and BBH, the axes of Figure 3. On the same sample, IFEval alone, BBH alone, a three-axis mean that adds GSM8K and a four-axis mean that also adds the multiple-choice suite put our rule at a worst-backbone percentile of 79, 69, 67 and 67, against the 81 reported here.

Over each backbone's contiguous 25% blocks, the summed probe correlates with IFEval after repair in the predicted direction on all five backbones, and strongly on three. It puts the best-recovering block in its top 5 on 3 of 5. On Qwen2-7B, whose best block is the shallow one, it ranks that block near the bottom. The probe alone points the right way but does not find the best block everywhere.

	Sets we ran			Ours	
Backbone	worst	median	best	score	pct.
Qwen3-8B	9.4	43.4	61.2	**64.2**	100%
Qwen2-7B	9.8	35.6	42.3	38.9	81%
Qwen1.5-7B-Chat	14.9	32.3	45.9	**46.7**	100%
Llama-3.1-8B	12.2	40.3	64.4	63.8	98%
Mistral-7B	18.3	38.7	54.0	51.0	93%

Table 10: How close our rule gets to the best set we tried, and how far apart the sets are. The worst, median and best score among the 285 contiguous-block and uniformly random sets (52–65 per backbone), and where ours lands. The score is the mean of IFEval (instruction-level strict accuracy) and BBH. Each of those sets is a single-seed run, while *Ours* is a three-seed mean, so the sampled sets are noisier. Figure 8 uses a wider collection that adds the overlap-stratified and one-layer-swap sets.

E.3 Do sets that look like a criterion's choice repair well?

Elsewhere we score what each criterion produced. Here we ask the reverse: over many removals cut and repaired the same way, do sets that share more layers with a criterion's choice repair better? The collection holds the criteria's own selections, random sets, contiguous blocks at every depth and two families that vary the overlap with our rule, so every degree of resemblance is covered. We correlate rather than compare against the single best set, because the sample leans towards contiguous blocks and would favour rules that cut them.

Result. On the three-axis mean, our rule has the highest correlation averaged over the five backbones (Figure 8, right). Per backbone, it is highest on Qwen3-8B and Qwen1.5-7B, the two backbones where the criteria disagree most. Block influence edges it on Llama-3.1-8B and, with angular distance, on Mistral-7B, the two backbones whose safe block is deep and contiguous. On Qwen2-7B, where every correlation is weak, DarwinLM leads, and it sits below ours on the other four backbones. Training on a recovery objective does not by itself make a criterion point to where repair helps. Angular distance points the wrong way on Qwen1.5-7B, where it removes the input stack at all three seeds (Figure 4).

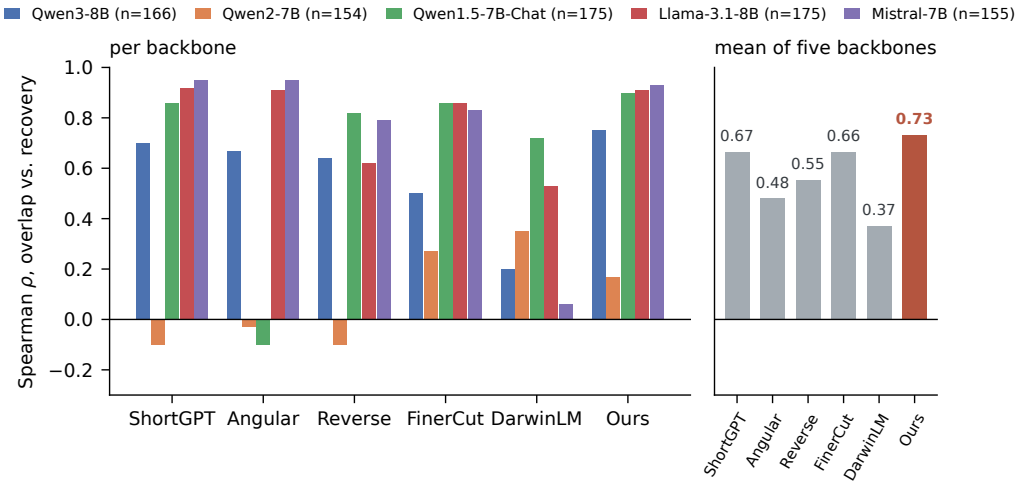


Figure 8: Do sets that resemble a criterion's choice repair well? *Left*: Spearman correlation between how many layers a set shares with a criterion's selection (averaged over its three seeds) and the set's three-axis mean share of the unpruned model after repair, one bar per backbone (n sets each, legend). *Right*: the mean over the five backbones. The axes are IFEval, GSM8K and BBH. Multiple choice barely moves across criteria and is left out.

E.4 More demonstrations, better prediction: the probe as a function of k

If the probe predicts recovery because it measures learning from demonstrations, its predictive power should grow with k and be weakest at k=0. Without new training, we re-measure every layer's probe at $k \in \{0, 1, 2, 4, 8\}$ on each backbone's contiguous blocks and random sets, the only sets of the map that no criterion chose and none was built around, sum it over each set as Eq. 3 does, and correlate the sum with IFEval after repair. Averaged over backbones, the correlation rises up to k=4 and eases after (Table 11).

The raw curve understates the effect. Without demonstrations no model answers the probe above chance (Appendix D), so the probe then measures only how far the removal moved the model and tracks block influence ($\rho = 0.91$, 0.93 and 0.92 on Llama-3.1-8B, Mistral-7B and Qwen1.5-7B). On these backbones present damage already predicts recovery, and the probe inherits that correlation without using a demonstration. Partialling out block influence isolates what the demonstrations add. On instruction following, the backbone mean is 0.02 at k=0, 0.26 at k=1, 0.33 at k=2 and 0.37 at k=4, the curve in Figure 3. On all 797 repaired sets the rise is smaller (IFEval $+0.23$ to $+0.34$, BBH $+0.42$ to $+0.57$ at k=2). Demonstrations also move the probe away from present damage where it matters. On Qwen3-8B its correlation with block influence falls from 0.82 at k=0 to 0.53 at k=2.

Where the rule's k comes from. The rule does not take its k from this curve. Each family uses the fewest demonstrations that define its task, one worked example for a stated rule and two input→output pairs for an induced one (one pair leaves the rule undetermined), and the admission gate is checked at the same k. For the induction families, k=2 sits just below the peak. With all seven families at that k, the correlation pooled over the same blocks and random sets is 0.70.

	induction families only, k					all seven families
Backbone	0	1	2	4	8	rule's k
Qwen3-8B	0.31	0.54	0.68	0.71	0.70	0.64
Qwen2-7B	0.16	0.13	0.17	0.40	0.34	0.46
Qwen1.5-7B	0.44	0.61	0.59	0.57	0.56	0.69
Llama-3.1-8B	0.65	0.71	0.71	0.69	0.68	0.73
Mistral-7B	0.64	0.66	0.71	0.69	0.66	0.72

Table 11: The raw correlations behind the right column of Figure 3: Spearman ρ between the summed single-layer probe and IFEval after repair, per backbone over its contiguous blocks and random sets. *Left*: the three induction families at k demonstrations. *Right*: all seven families at the k the rule uses (Appendix D).

E.5 The recovery map under interleaved repair

175 of the map's contiguous blocks and random sets are also repaired on the interleaved schedule, with 200 steps after every removal at the same total budget. The ordering stays the same. Pooled over those sets, the summed probe leads both present-damage signals under either schedule and on both axes, with about the same margin over block influence, and it leads block influence on every backbone and axis. It leads the output divergence on 4 of the five backbones and trails it on Qwen2-7B in 3 of the 4 schedule–axis pairs. Scored on IFEval, the landscape keeps its two shapes, and the peaks move by a few layers.

E.6 The recovery map scored on BBH

Scoring the same runs on post-repair BBH instead of instruction following gives the bottom row of Figure 3. Pooled over sets, the probe leads block influence, and it also leads on every backbone taken alone (0.86 against 0.55 on Qwen3-8B, 0.50 against -0.06 on Qwen2-7B, 0.78 against 0.65 on Qwen1.5-7B, and 0.92 against 0.85 and 0.86 on Llama-3.1-8B and Mistral-7B). The landscape keeps its two shapes on this axis. The best block starts deep on the four deep-tolerant backbones and at layer 3 on the shallow-tolerant one.

E.7 Scope of the damage–recovery comparison

The comparison in §4.2 is made at a fixed repair budget, and Appendix I repeats the criteria at half and double it. Set-level d_0 saturates when a quarter of the layers goes at once. Neither signal identifies the best set everywhere. And which set is best after repair depends on the metric. Both selectors choose without seeing a metric, so we score them on both axes (Figure 3 and Appendix E.6).

F Pool source and probe role

What the pool does. The pool filters the candidates, and the probe chooses among them. In Table 14, every probe-selected variant keeps more on its worst backbone than the four baselines. On Llama-3.1-8B and Mistral-7B the probe picks the same layers with the block-influence pool as with none. On the three Qwen models the pool changes the set. On Qwen3-8B the probe alone scores the penultimate layer as cheap and removes it. Repair does not bring that layer's function back, and instruction following and BBH fall by about 7 points. The block-influence pool prevents this loss by never offering that layer. The loss is output-side damage that the probe cannot see (§3.1). On Qwen1.5-7B it swaps one layer and gains a few points. On Qwen2-7B the pool excludes two layers the probe would have taken and costs 2–4 points. Qwen2-7B is the weakest backbone and sets the worst case, so on the worst case the probe alone is ahead of the pooled rule. On the mean the pooled rule is ahead, if only by a point or two. A pool built on d_0, the JSD FinerCut optimises, does about as well. Inside it the probe changes 1–4 of the layers the JSD ranking would remove, on every backbone, and lifts the mean well above that ranking alone.

Why the rule keeps the pool. We keep the pool because the Qwen3-8B failure is the kind the probe cannot see. The pool also halves the probe's cost, since the probe runs on $2T$ candidates instead of every layer. Ranking each pool by its own present-damage score instead of by the probe shows where the probe is worth most. On the mean, it adds most where the base criterion is weakest (Table 14).

Pool width. The pool width is the rule's only hyper-parameter. We set it to $2T$, two candidates per removal slot on every backbone and seed, and did not tune it. From 50% removal on, $2T$ reaches the number of layers, so the pool holds every layer and the probe alone decides. On the vision–language models the width can be swept without training, because the accumulated schedule selects entirely on the never-repaired model. From $T/2$ to $2T$ per tower, the 2B split never moves, and the 8B split moves once ($7/1$ to $6/2$) and then settles.

Backbone	MCQ_{12}	IFEval	GSM8K	BBH
Qwen3-8B	**+1.4**	**+2.9**	**+9.5**	**+11.5**
Qwen2-7B	**+0.1**	**+4.7**	**+3.6**	**+4.0**
Qwen1.5-7B	**+0.8**	**+3.7**	**+1.6**	**+2.6**
Llama-3.1-8B	−0.4	**+1.9**	−1.1	**+0.3**
Mistral-7B	**+0.7**	−2.4	**+1.3**	−1.6
Mean	+0.5	+2.1	+3.0	+3.3

Table 12: The probe's own contribution at 25% removal: our rule minus the same pool ranked by block influence, three seeds, accuracy points on the axes of Table 2. Dropping the probe from Eq. 3 leaves $\mathcal{L}_T(\text{BI})$, so each entry is the Ours row of that table minus its ShortGPT row. The gain is concentrated on the reasoning axes and on the three Qwen backbones. It is small on Llama-3.1-8B and Mistral-7B, whose safe block is deep and contiguous. Other rates: Table 13.

G Rate curves and the tables at the other removal rates

These are the single-pass runs of §4.3 at the other rates, under the same rule and repair (Appendix B). Table 16 gives the 37.5 and 50% cells. On the worst backbone's three-axis mean, FinerCut is the strongest present-damage rule we ran, so §4.3 compares ours with it across rates (Table 16). Appendix I.2 asks whether any of this depends on the exemplars.

		MCQ_{12}		IFEval		GSM8K		BBH	
Removed	Rule	mean	worst	mean	worst	mean	worst	mean	worst
12.5% (n=5)	ShortGPT	99.8	90.4	93.1	79.1	78.2	52.3	97.6	54.0
	Ours (ShortGPT + ICL probe)	99.5	90.7	91.9	78.7	79.0	58.2	98.8	57.8
	Δ	−0.3	+0.3	−1.2	−0.4	+0.8	+5.9	+1.2	+3.8
25% (n=5)	ShortGPT	92.0	83.5	81.7	68.5	55.8	32.6	84.5	43.3
	Ours (ShortGPT + ICL probe)	92.9	83.6	84.9	73.9	59.7	37.3	90.1	49.9
	Δ	+0.9	+0.1	+3.3	+5.3	+3.9	+4.7	+5.6	+6.6
37.5% (n=5)	ShortGPT	84.3	74.8	65.9	51.0	30.9	18.0	57.2	26.1
	Ours (ShortGPT + ICL probe)	84.2	77.2	71.2	52.6	34.6	15.1	64.1	26.4
	Δ	0.0	+2.5	+5.3	+1.6	+3.7	−2.9	+7.0	+0.3
50% (n=5)	ShortGPT	74.0	70.3	52.1	40.9	10.4	4.5	30.9	11.4
	Ours (ShortGPT + ICL probe)	75.8	69.6	55.0	36.5	16.0	6.4	46.0	22.7
	Δ	+1.8	−0.7	+2.9	−4.4	+5.7	+2.0	+15.0	+11.3

Table 13: The probe's contribution at each removal rate, on the four axes of Table 2: share of the unpruned model, averaged over the five backbones (*mean*) and at the worst of them (*worst*), three seeds per cell. Δ is Ours minus ShortGPT, the probe ablation. The probe adds BBH and GSM8K on the mean at every rate, most at 50%. The mean includes Qwen1.5-7B, whose BBH share exceeds 100% because the repair data lifts BBH above the unpruned model (Appendix K).

Variant	Qwen3	Qwen2	Qwen1.5	Llama-3.1	Mistral	worst	mean
ShortGPT	68%	56%	118%	82%	92%	56%	83%
Angular	73%	57%	58%	84%	92%	57%	73%
Reverse order	64%	33%	90%	68%	80%	33%	67%
FinerCut (d_0 only, no probe)	62%	60%	117%	76%	84%	60%	80%
Ours (BI pool + probe selects)	77%	63%	126%	83%	89%	63%	**88%**
BI pool + DarwinLM	48%	59%	101%	66%	75%	48%	70%
Angular pool + probe selects	69%	67%	123%	83%	89%	67%	86%
d_0 (JSD) pool + probe selects	72%	63%	124%	83%	92%	63%	87%
Probe alone (no pool)	69%	68%	123%	83%	89%	**68%**	86%

Table 14: Pool source and probe role: share of the unpruned model retained at 25% (the mean of the IFEval and BBH shares), worst and mean over five backbones, three seeds (DarwinLM: one). Shares above 100% on Qwen1.5-7B: Appendix K. Rows differ only in where the candidates come from and what ranks them. Bold: the best worst case and the best mean.

Backbone	Probe data	MCQ_{12}	IFEval	GSM8K	BBH
Qwen3-8B	Synthetic suite (default)	**58.3**	**65.3**	**60.1**	**63.1**
	ICLEval, generative ICL tasks	**58.3**	65.2	**60.1**	**63.1**
Qwen2-7B	Synthetic suite (default)	51.4	**47.3**	**28.7**	30.4
	ICLEval, generative ICL tasks	**52.1**	38.4	18.7	**33.2**
Qwen1.5-7B	Synthetic suite (default)	**54.1**	**51.6**	30.5	**41.8**
	ICLEval, generative ICL tasks	53.4	49.3	**33.1**	39.1
Llama-3.1-8B	Synthetic suite (default)	**59.6**	66.7	50.4	**60.9**
	ICLEval, generative ICL tasks	**59.6**	**67.8**	**52.0**	60.4
Mistral-7B	Synthetic suite (default)	**56.9**	54.2	**36.8**	47.9
	ICLEval, generative ICL tasks	56.2	**59.4**	31.6	**49.0**
Mean	Synthetic suite (default)	**56.1**	**57.0**	**41.3**	48.8
	ICLEval, generative ICL tasks	55.9	56.0	39.1	**49.0**

Table 15: **The same rule with two probe corpora.** Our synthetic suite (the default) against the public generative tasks of ICLEval (Chen et al., 2025b), at 25% removal. The ICLEval task built from GSM8K test questions is excluded. On the five-backbone mean the two land close together, with the largest gap on GSM8K, while single backbones move by several points. With these six tasks, the synthetic suite is a convenience rather than an essential part of the method. Bold: the better corpus in each column of a block.

		37.5% removed				50% removed			
Backbone	Criterion	MCQ_{12}	IFEval	GSM8K	BBH	MCQ_{12}	IFEval	GSM8K	BBH
Qwen3-8B	Unpruned	56.0	88.4	86.4	78.6	56.0	88.4	86.4	78.6
	ShortGPT	**48.9**	45.0	18.8	20.5	43.4	**36.1**	4.9	9.0
	Angular	45.5	38.4	7.4	15.3	41.6	26.3	2.0	2.8
	Reverse order	44.1	26.9	3.3	15.6	39.9	23.1	1.8	4.1
	FinerCut	**48.9**	44.8	**25.7**	**25.0**	**44.7**	35.3	**12.1**	8.7
	Ours	48.1	**46.5**	13.0	20.7	43.3	32.3	5.6	**17.9**
Qwen2-7B	Unpruned	61.5	62.2	76.8	60.9	61.5	62.2	76.8	60.9
	ShortGPT	46.0	35.0	13.8	21.5	43.2	29.5	3.4	12.8
	Angular	47.2	37.5	12.4	18.6	43.6	28.7	4.5	14.5
	Reverse order	40.2	19.7	2.1	12.6	38.8	19.9	1.8	5.6
	FinerCut	**48.1**	**43.7**	16.4	20.4	**43.7**	29.1	4.7	14.4
	Ours	47.5	42.2	**18.4**	**27.6**	42.8	**33.4**	**10.4**	**19.2**
Qwen1.5-7B	Unpruned	56.3	50.6	57.6	27.9	56.3	50.6	57.6	27.9
	ShortGPT	**50.3**	37.1	15.3	24.2	40.8	28.0	2.7	11.0
	Angular	42.1	28.1	3.8	6.5	38.6	26.9	2.6	8.3
	Reverse order	47.0	29.4	5.7	14.2	39.1	21.6	2.1	8.8
	FinerCut	49.3	43.8	**27.5**	29.1	44.1	**36.7**	14.3	18.1
	Ours	**50.3**	**46.1**	23.5	**31.8**	**47.7**	35.4	**14.8**	**27.5**
Llama-3.1-8B	Unpruned	64.8	81.8	76.0	72.1	64.8	81.8	76.0	72.1
	ShortGPT	**56.3**	55.3	33.4	**49.4**	48.8	38.9	11.2	26.1
	Angular	44.6	32.1	4.6	5.9	41.5	27.7	2.8	3.0
	Reverse order	54.9	40.2	33.1	35.8	46.5	30.9	5.8	17.5
	FinerCut	56.2	**56.7**	**40.2**	42.5	**49.1**	**46.1**	**19.9**	27.6
	Ours	56.2	55.0	36.7	48.2	48.5	41.0	10.7	**30.7**
Mistral-7B	Unpruned	64.2	59.4	51.0	55.7	64.2	59.4	51.0	55.7
	ShortGPT	**53.2**	48.3	22.5	38.6	**48.0**	**41.0**	11.4	**26.0**
	Angular	53.1	**48.8**	20.8	**40.0**	44.5	36.3	7.3	16.0
	Reverse order	52.3	41.5	18.3	32.9	44.2	28.9	5.4	10.2
	FinerCut	49.2	45.7	**23.4**	26.7	46.3	39.4	**15.6**	15.9
	Ours	52.5	45.9	22.9	38.0	46.4	38.3	10.5	19.2

Table 16: The single-pass protocol at 37.5 and 50% removal, accuracy (%), means of three seeds, on the four axes of Table 2. The unpruned row is the same checkpoint at both rates. Bold: best of the five criteria within a backbone at that rate. The two reasoning axes diverge here. Ours leads BBH on the five-backbone mean at both rates, while FinerCut leads GSM8K (most of the lead is on Qwen3-8B and Llama-3.1-8B) and has the best instruction following on two backbones at each rate.

G.1 What a training-aware criterion selects

The criteria of the main text score a layer by the damage its removal does *now*. A method that trains a little before scoring might already capture what our probe captures. We port DarwinLM's training-aware idea to our setting and run it under our protocol. Its accuracies are a row of Table 2 under both repair objectives. This subsection asks the prior question of *which layers* it picks.

Each round removes the layer whose removal leaves the smallest $\mathrm{KL}(\text{dense} \,\|\, \text{sparse})$ on gold response tokens. The KL is measured *after* a short in-loop supervised pass (160 steps at batch 16, 2,560 samples) on the dense model's own greedy responses, so that the training target and the KL reference agree. Two caveats apply. First, DarwinLM as published searches non-uniform *structured* sparsity (attention heads and MLP width, per module) by evolution, whereas ours is a greedy *depth* port, the only form comparable to the rest of this paper. Second, the in-loop data (the dense model's own responses instead of DarwinLM's web text) and budget are our choices rather than the method's, so this baseline tests a short training pass, not the best a training-aware criterion could do.

A short in-loop pass does not move the criterion away from damage. On every backbone, DarwinLM's sets share more of their layers with the damage-only baseline than with ours, 58% against 36% averaged over backbones and seeds. Layer by layer (Figure 4), DarwinLM's row sits nearest

FinerCut's, and on four of the five backbones it is farther from the best 3 runs than ours is. Restricted to our block-influence pool, the same selector still lands far below the probe's choice (Table 14, one seed).

H When to score: once, or after each removal

Three schedules. The *single pass* scores the pool once on the original model and removes the budget at once. *Accumulated* marks the cheapest layer as removed, re-scores the survivors without it and repeats until the budget is spent. It repairs only at the end. *Iterative* also repairs after each removal. The total repair budget is the same in all three, so they differ only in when the scores are measured (Appendix B.2). Our rule uses the single pass on one stack and accumulates on two towers, as Observation 2 predicts. Table 17 puts the three side by side on both architectures. Per-axis values are in Appendix N (Tables 33 and 34).

	Single pass *score once, on the original model*	Accumulated *re-score after each removal*	Iterative *re-score and repair after each removal*
Five language models (25% removal; mean of IFEval, GSM8K and BBH)			
Qwen3-8B	**62.8**†	59.7	54.6
Qwen2-7B	35.5†	**37.4**	35.3
Qwen1.5-7B	**41.3**†	36.7	32.3
Llama-3.1-8B	**59.3**†	54.1	54.0
Mistral-7B	**46.3**†	43.0	38.8
Mean of the five	**49.0**†	46.2	43.0
Four vision–language models (eight layers; average of the four axes)			
InternVL3.5-2B	45.7	**47.0**†	45.2
InternVL3.5-8B	56.0	**56.5**†	56.0
Qwen2-VL-2B	40.3	**41.3**†	37.8
Qwen2-VL-7B	48.7	**49.5**†	46.6

Table 17: **The three schedules on one stack and on two towers.** Score under each schedule. † marks our rule's schedule, and bold the best in the row. Language models: 25% removal, three seeds, mean of IFEval, GSM8K and BBH (MCQ_{12} barely moves between schedules). Vision–language models: eight layers, average of the four axes, all three schedules over the same pool and probe (InternVL3.5 three seeds, Qwen2-VL one). Twelve and sixteen layers: Table 34.

One stack: score once. The single pass beats accumulating on 4 of the five backbones and beats the iterative variant on all 5 (Table 17). Per axis, accumulating loses on both pools on every axis of the mean, except instruction following on the angular pool. It loses most on BBH, where the probe helps most (Table 33), and repairing between removals does not rescue it. The angular pool is the harder case for us, because re-measuring should help a contiguous-block rule as the stack shrinks, yet it does not. As in Observation 2, re-scoring measures each survivor on the unrepaired network, a model that will never be used without repair. When the last layers are removed, the remaining deep layers look ever more costly, and the rule moves to middle layers (Appendix E.1).

Two towers: accumulate. Here the comparison runs the other way (Table 17). Accumulating beats the single pass over the same pool and probe on every backbone, and on the average at every budget of Table 34, most on long-form generation and grounding. Repairing between removals does not help either. The iterative schedule is below the accumulated rule on all four backbones, mostly on text reading and long-form generation.

Where the two-tower schedules differ. At 8 layers the single pass and accumulation split the budget between the towers alike, to within one layer, so what separates them is mostly placement inside the vision tower. The single pass removes vision layers 16–22 as one contiguous run, on 2B in every seed and on 8B in two of three. Accumulation spreads its vision removals: layers 2, 7, 13, 16, 17 and 20–22 on 2B, and 5, 16 and 18–22 on 8B. The single pass keeps text reading but loses

long-form generation and grounding, by several points on 2B and about a point on 8B (Table 34, 8-layer rows). At 12 and 16 layers it also takes up to two more vision layers than accumulation does.

Why accumulation wins on two towers. Both effects have one cause. The vision tower's costs rise fastest as the budget is spent, and a single pass scores every vision layer as if it were the first to go. On the accumulated rule's own never-repaired model, the cheapest vision candidate starts at 0.29 to 0.47 of the cost of the cheapest language one, across the four backbones. Over the 8 removals its cost then rises 5.5 to 12.9 times, against the language tower's 2.4 to 4.6, about 3 times as fast. On the larger pair the two costs cross before the budget is spent (InternVL3.5-8B at the seventh removal, Qwen2-VL-7B at the eighth), but not on the 2B pair. On two towers re-scoring is informative, and it needs only probe passes on the never-repaired model.

I Does the ordering survive more repair, or a different prompt?

I.1 More repair

Table 18 asks whether the ordering of §4.3 depends on the repair budget. The five criteria's 25% removal sets from Table 2 are repaired again, unchanged, at half and at double the budget of §4.1, with the same recipe and learning rate and one seed at all three. Table 19 gives the same runs per backbone.

	IFEval			GSM8K			BBH		
Criterion	half	**as used**	double	half	**as used**	double	half	**as used**	double
ShortGPT	54.2	55.0	56.5	34.0	38.4	41.9	45.3	45.5	45.0
Angular	54.2	54.2	55.3	31.6	35.2	39.1	40.5	41.0	42.0
Reverse	40.4	42.0	46.7	30.3	33.0	37.4	38.0	38.7	40.3
FinerCut	56.1	55.6	**59.0**	38.9	41.5	44.4	39.8	40.6	40.5
Ours	**57.3**	**57.4**	57.6	**39.3**	**42.2**	**45.7**	**47.8**	**48.9**	**49.2**

Table 18: Halving and doubling the repair at 25% removal, accuracy ($\%$) averaged over the five backbones. *As used* is the budget of §4.1, 200 optimiser steps per removed layer. The sets are those of Table 2 and do not change, so a column moves only through repair. Bold: best of the five criteria. Per-backbone values: Table 19 below.

On the two reasoning axes the ordering survives, and the margin grows with the budget. On BBH ours is the best criterion at every budget, with a widening lead and the best worst-backbone score at all three. GSM8K has the same ordering with a smaller lead. Instruction following differs. Ours leads at the two smaller budgets, but at the largest FinerCut passes it, because the doubled repair lifts the baselines' IFEval and barely moves ours. On that axis, extra repair gives the baselines what our selection had already achieved.

I.2 A different prompt: BBH without exemplars

Every table reports BBH with three exemplars, the protocol of prior depth-pruning work. This choice could matter, because exemplars show the answer format, so a repair that teaches a format could be rewarded for formatting rather than reasoning. Table 20 therefore also scores the same checkpoints zero-shot, with an answer extractor of its own applied identically to every model.

The protocol does not change the conclusion. Within a backbone, the two protocols rank the six criteria almost identically (mean Spearman ρ of 0.87), and ours has the best mean and the best worst case under both (Table 20). Our margin over the best baseline is smaller without exemplars on every backbone, and on Qwen2-7B DarwinLM edges ahead of ours without them. Among the unpruned models, only Qwen1.5-7B scores higher without exemplars (Appendix K). We report the three-exemplar protocol in the body because the baselines were published under it.

		IFEval			GSM8K			BBH		
Backbone	Criterion	half	**as used**	double	half	**as used**	double	half	**as used**	double
Qwen3-8B	ShortGPT	62.0	61.5	63.8	46.8	48.4	51.9	50.3	52.0	44.8
	Angular	64.3	64.3	63.2	37.5	42.5	46.8	56.5	58.3	59.5
	Reverse	50.5	50.8	55.4	47.8	51.6	52.9	51.4	54.2	54.3
	FinerCut	60.8	61.0	64.3	49.1	53.9	54.2	42.4	43.7	43.5
	Ours	**64.9**	**67.5**	**68.3**	**58.5**	**60.8**	**61.8**	**60.0**	**62.9**	**61.9**
Qwen2-7B	ShortGPT	40.4	43.6	44.4	21.2	25.4	27.2	27.2	26.0	28.3
	Angular	45.4	45.0	45.1	20.5	22.7	25.2	26.9	25.9	26.9
	Reverse	24.0	26.1	27.8	2.8	2.8	6.7	13.2	12.5	13.7
	FinerCut	47.6	**47.5**	**48.7**	**26.8**	**30.9**	**35.0**	28.8	29.5	26.9
	Ours	**48.1**	47.2	46.0	26.0	29.9	32.2	**30.4**	**31.0**	**31.5**
Qwen1.5-7B	ShortGPT	47.2	46.0	47.2	21.8	28.2	32.5	38.6	39.9	40.7
	Angular	35.4	38.1	39.1	15.5	18.9	22.9	7.9	9.6	10.8
	Reverse	37.2	37.3	42.6	22.6	26.8	32.2	27.4	29.1	32.1
	FinerCut	**53.1**	**52.0**	**55.5**	**37.0**	**37.7**	**42.2**	36.3	37.2	37.7
	Ours	52.6	51.7	50.4	26.5	29.8	37.8	**40.0**	**41.9**	**41.8**
Llama-3.1-8B	ShortGPT	65.3	66.3	66.8	52.1	52.4	57.8	60.5	60.3	61.3
	Angular	**69.8**	66.3	**68.5**	**56.2**	**54.3**	**61.3**	**61.7**	**62.0**	**62.4**
	Reverse	47.1	48.9	54.9	47.5	49.0	54.7	53.3	53.0	54.9
	FinerCut	66.3	63.8	68.0	49.7	50.1	52.3	51.2	52.0	52.8
	Ours	69.1	**67.6**	67.3	51.8	50.8	55.6	61.0	60.9	61.2
Mistral-7B	ShortGPT	**56.2**	**57.3**	60.3	28.2	37.8	39.9	**49.8**	**49.3**	**50.1**
	Angular	56.1	**57.3**	**60.4**	28.2	37.8	39.5	49.7	**49.3**	**50.1**
	Reverse	43.3	47.0	52.6	30.7	35.0	40.3	44.9	44.7	46.7
	FinerCut	52.5	53.8	58.4	31.8	34.9	38.1	40.4	40.8	41.6
	Ours	51.9	53.1	56.2	**33.7**	**39.7**	**40.8**	47.9	47.8	49.5

Table 19: The same runs per backbone: each criterion's 25% removal set repaired at half, at the budget of §4.1 and at double it. Accuracy (%), one seed and learning rate 10^{-4} throughout. The sets are fixed, so a row moves only through repair. Bold: best of the five criteria within a backbone, axis and budget. ShortGPT is also the probe ablation, so the Ours–ShortGPT gap at each budget is what the probe is worth at that budget.

	Zero-shot		Three exemplars	
Criterion	mean	worst	mean	worst
Unpruned	53.2	41.4	59.0	27.9
ShortGPT	36.1	26.3	45.5	26.4
Angular	32.0	14.2	41.7	11.2
Reverse order	33.5	17.6	39.3	13.8
FinerCut	32.2	25.2	40.2	27.9
DarwinLM	28.9	23.3	32.5	28.6
Ours	**37.2**	**29.4**	**48.8**	**30.4**

Table 20: BBH at 25% removal scored without exemplars and with the three exemplars of the body: mean and worst over the five backbones of Table 2, three-seed means (DarwinLM's zero-shot scores on Llama-3.1-8B and Mistral-7B: two seeds). The unpruned row is a reference and is never bold. Its worst three-exemplar score is Qwen1.5-7B's (Appendix K). Bold: best criterion in the column.

J CHANGING THE REPAIR OBJECTIVE: ON-POLICY DISTILLATION

Table 18 asks whether the ordering survives more of the same repair. This section asks whether it survives a different repair. Only the loss changes, and everything else is as in §4.1, including the $200T$-step budget. Supervised fine-tuning raises the likelihood of a gold continuation. The substitute, on-policy distillation (Agarwal et al., 2024), lets the *pruned* model generate a response to the same prompt. It then minimises the Jensen–Shannon divergence to the unpruned model's next-token distribution at those generated positions. The rollouts are regenerated every 200 steps, T times over the budget. The teacher is the unpruned model throughout, and no gold continuation is

used. The right half of Table 2 runs the substitute on all five backbones. Table 21 puts supervised fine-tuning, distillation and the two in series side by side on Qwen3-8B.

The ordering depends on the selection rather than on the repair. §4.3 counts the column wins. What the objective changes is the balance of axes, and this change depends far more on the backbone than on the criterion. On the five-backbone mean, multiple choice falls on all six criteria. This is what a model trained on its own continuations, and never shown the multiple-choice answer format, should do. Instruction following rises on every criterion on Qwen3-8B and falls on every criterion on Mistral-7B.

Running the two in series trades BBH for the other three axes. The series variant spends the fine-tuning budget and then the distillation budget, twice the budget of either alone. Against distillation alone, it is higher on multiple choice, instruction following and GSM8K for every criterion, ours included, and lower on BBH for every criterion. It keeps little of fine-tuning's multiple choice, and on BBH it is never the best of the three. The gains of the two objectives do not add.

Whether the substitute helps depends on what the teacher knows. GSM8K moves in both directions across the backbones of Table 2, and the direction follows the unpruned model's own accuracy on that benchmark. Across the five backbones, the teacher's GSM8K and the mean gain of distillation over supervised fine-tuning correlate at $r = 0.93$. The substitute gains most over supervised fine-tuning on Qwen3-8B, whose teacher is the strongest of the five. It loses most on Mistral-7B, whose teacher is the weakest. This follows from the objective. On-policy distillation never uses a gold continuation, so its only signal about a correct chain of arithmetic is the teacher's own distribution. A teacher that solves half the problems supplies a signal that is half wrong. Supervised fine-tuning uses gold answers and does not depend on the teacher.

The GSM8K gap is not a scoring artefact. Distilled models write longer answers, so we also scored Mistral-7B, where the gap is largest, with a lenient bound that accepts the gold number anywhere in the output. Distillation stays well behind, so the answer parser does not change the conclusion.

Distillation needs a student close to its teacher. We ran the same comparison at 12.5, 37.5 and 50% removal on all five backbones and criteria (Figure 9). The instruction-following gain that motivates this comparison appears at 25% and largely on Qwen3-8B. Beyond 25%, distillation loses on every axis of the five-backbone mean, and at 50% it trails fine-tuning in 71 of the 75 backbone and criterion cells of the three generative axes, by 4.5 to 7.8 points on the mean. We read this as a gap between student and teacher. Distillation trains only on the positions the pruned model itself generates and pulls its distribution there towards the teacher's. The more layers are removed, the further these rollouts move from anything the teacher would write, and the teacher's distribution at them says little about how to recover. Gold continuations give the same signal whatever the student's state. Supervised fine-tuning is therefore the repair everywhere else in the paper.

Data seen by each objective. Optimiser steps, batch, corpus and removal sets are identical between the two objectives, but the data behind those steps is not. A distillation run reuses each rollout about 6 times to fill the same number of sample presentations. It therefore sees roughly 10 times fewer distinct examples than supervised fine-tuning. In exchange, it supervises about twice as many positions per example and gives each position the teacher's full distribution rather than one label.

K What the repair data does to an unpruned model

Table 22 answers why a pruned model can score above the unpruned one. It repairs the unpruned model on the same data with nothing removed, which separates what the repair data does from what the removal costs. The case that needs it is BBH on Qwen1.5-7B, the oldest of the five backbones. There the repair data alone lifts three-exemplar BBH far above the unpruned model, and removal then takes back only a few points, so under fine-tuning 5 of the 6 pruned models score above the unpruned model (Table 2). Its unpruned three-exemplar score is also unusually low. It is the only backbone whose unpruned BBH rises when the exemplars are dropped (Appendix I.2). Together,

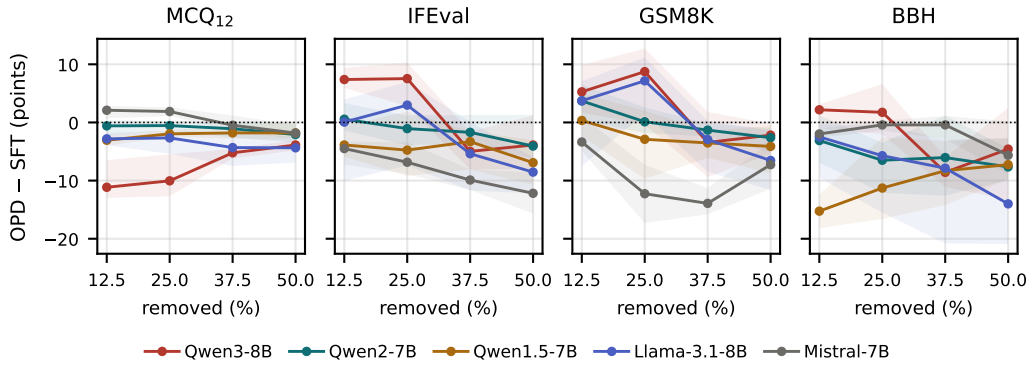


Figure 9: Distillation minus supervised fine-tuning, in accuracy points, against the fraction of layers removed, one panel per axis of Table 2. Each line is one backbone and shows the mean over the five criteria, with their range as a band. Beyond 25% removal the five-backbone mean is below zero on every axis. Outside 25%, distillation is one seed against three for supervised fine-tuning.

	SFT				OPD				SFT, then OPD			
Criterion	MCQ_{12}	IFEval	GSM8K	BBH	MCQ_{12}	IFEval	GSM8K	BBH	MCQ_{12}	IFEval	GSM8K	BBH
ShortGPT	**56.9**	62.4	50.6	**51.7**	46.8	70.4	62.4	49.8	48.7	**72.5**	**65.4**	48.7
Angular	**57.8**	62.9	44.3	59.3	45.1	72.0	43.9	**62.1**	46.6	**73.1**	**54.0**	58.2
Reverse order	**55.6**	51.5	51.9	54.0	46.1	54.3	64.6	**60.6**	47.3	**65.3**	**68.5**	54.1
FinerCut	**57.4**	60.9	52.7	**43.9**	51.9	68.4	63.8	41.9	52.6	**71.5**	**66.4**	41.6
Ours	**58.3**	65.3	60.1	63.1	45.9	75.6	68.6	**66.3**	47.7	**77.1**	**72.1**	61.0

Table 21: Supervised fine-tuning, on-policy distillation, and the two in series (SFT, then OPD) on Qwen3-8B at 25% removal, accuracy (%), means of three seeds. SFT and OPD each spend the repair budget of Table 2 and repeat its Qwen3-8B block. The series spends both back to back, twice either alone. Bold: the best of the three objectives for each criterion and axis.

these are why Qwen1.5-7B's BBH share exceeds 100% in the tables of shares. On the other backbones the data moves BBH by a few points at most. It moves the model more than the removal does in only two other cells: multiple choice on Qwen3-8B, whose unpruned model handles the letter format poorly, and instruction following on Qwen1.5-7B.

The same reading for the vision–language backbones. Table 23 repeats the comparison for InternVL3.5 on the four columns of Table 3. Here the repair data is close to neutral on multiple choice and text reading but costs open-ended generation on both backbones. That is the only axis where the unpruned model's long answers are worth more than the repair mixture's short ones. Removal then lowers every axis further. Under the data alone, grounding moves in opposite directions on the two sizes, so its share of the unpruned model is the least clean of the four.

K.1 A KNOWLEDGE-DOMINATED BACKBONE

Qwen1.5-7B-Chat is the most knowledge-dominated of the five backbones, and the criteria do not agree on it. Ours is best on 3 of the four columns of Table 2, and the BBH spread across the published criteria is the widest of any backbone. The boundary this backbone marks lies in the repair data, not in the criterion. As shown above, its unpruned BBH rises without exemplars and the repair data alone lifts it sharply, so its reasoning axis is measured on a narrower and more movable base than the others'.

Backbone	Model	MCQ_{12}	IFEval	GSM8K	BBH
Qwen3-8B	Unpruned	56.0	88.4	86.4	78.6
	Unpruned + SFT	67.2	80.1	83.2	77.0
	Δ repair data	$+11.2$	-8.3	-3.2	-1.6
	Ours (25% removed, SFT)	58.3	65.3	60.1	63.1
	Δ removal	-8.9	-14.8	-23.1	-13.9
Qwen2-7B	Unpruned	61.5	62.2	76.8	60.9
	Unpruned + SFT	62.8	62.9	77.4	59.5
	Δ repair data	$+1.3$	$+0.7$	$+0.6$	-1.4
	Ours (25% removed, SFT)	51.4	47.3	28.7	30.4
	Δ removal	-11.4	-15.6	-48.7	-29.1
Qwen1.5-7B	Unpruned	56.3	50.6	57.6	27.9
	Unpruned + SFT	58.3	55.6	54.4	44.1
	Δ repair data	$+2.0$	$+5.0$	-3.2	$+16.2$
	Ours (25% removed, SFT)	54.1	51.6	30.5	41.8
	Δ removal	-4.2	-4.0	-23.9	-2.3
Llama-3.1-8B	Unpruned	64.8	81.8	76.0	72.1
	Unpruned + SFT	65.8	76.2	75.0	71.3
	Δ repair data	$+1.0$	-5.6	-1.0	-0.8
	Ours (25% removed, SFT)	59.6	66.7	50.4	60.9
	Δ removal	-6.2	-9.5	-24.6	-10.4
Mistral-7B	Unpruned	64.2	59.4	51.0	55.7
	Unpruned + SFT	61.3	64.8	51.0	57.8
	Δ repair data	-2.9	$+5.4$	$+0.0$	$+2.1$
	Ours (25% removed, SFT)	56.9	54.2	36.8	47.9
	Δ removal	-4.4	-10.6	-14.2	-9.9

Table 22: The unpruned model, the unpruned model after the same repair as the pruned models (the same data and one $200T$-step LoRA run, nothing removed), and our method after the same repair with 25% removed, on the four axes of Table 2; accuracy (%). Δ repair data = (Unpruned+SFT) − Unpruned, and Δ removal = Ours − (Unpruned+SFT). Unpruned is one run. Unpruned+SFT and Ours are three-seed means, Ours as in Table 2.

Backbone	Model	MCQ_8	$Text_4$	$Long_2$	$Ground_2$
InternVL3.5-2B	Unpruned	64.3	75.6	68.0	24.7
	Unpruned + SFT	63.6	75.0	56.3	26.7
	Δ repair data	-0.7	-0.6	-11.7	$+2.0$
	Ours (eight layers removed, SFT)	58.2	62.1	44.4	23.2
	Δ removal	-5.4	-12.9	-11.9	-3.5
InternVL3.5-8B	Unpruned	70.6	79.6	74.9	38.3
	Unpruned + SFT	70.7	79.4	62.7	34.8
	Δ repair data	$+0.1$	-0.2	-12.2	-3.5
	Ours (eight layers removed, SFT)	67.6	74.0	51.6	32.7
	Δ removal	-3.1	-5.4	-11.1	-2.1

Table 23: The unpruned vision–language model, the same model after the pruned models' repair (768 LoRA steps on the same data, nothing removed), and our method after the same repair with 8 layers removed, on the four columns of Table 3; accuracy (%). Δ repair data = (Unpruned+SFT) − Unpruned, and Δ removal = Ours − (Unpruned+SFT). Unpruned is one run. Unpruned+SFT is the mean of three seeds, and Ours the three-seed mean of Table 3.

L VISION–LANGUAGE EXPERIMENTS

L.1 SETUP

Backbones and budget. InternVL3.5-2B and InternVL3.5-8B (Wang et al., 2025), 8 layers removed by every criterion, with one 768-step repair after the cut, three seeds. The harder budgets of Table 25 scale the repair with the budget.

The pool. Unlike in the language-only runs, every vision and every language block is a candidate, and the criterion decides the split. Two baselines fix the split instead: Reverse (ViT) takes the deepest vision blocks, and Reverse (LM) the deepest language blocks. Block influence and angular distance rank the same two-tower pool as ours. For ours the pool holds the T cheapest units of each tower, $2T$ in all as in §3.2, and the probe is the TrueMICL suite of Appendix D alone, at k=3.

Benchmarks. Sixteen sets, all through lmms-eval (Zhang et al., 2025), grouped by answer format so that no axis rests on a single scorer. MCQ_8 averages MMStar (Chen et al., 2024) with seven letter-choice sets. On BLINK the question is followed by "Answer with the option letter only." Text_4 averages OCRBench (Liu et al., 2024) with ChartQA, TextVQA and InfoVQA. Long_2 averages the two GPT-judged open-ended sets, MMVet and LLaVA-Bench-in-the-Wild. Grounding averages box-centre pointing on 966 PointArena items and free-space pointing on 100 Where2Place items. Per-set scores are in Appendix N.

Repair, selection and cost. LoRA rank 8, α=16, learning rate 10^{-4}, global batch 64, 768 optimiser steps per criterion, on a single-image subset of the MAmmoTH-VL instruction data (Guo et al., 2025) (49,152 examples drawn per run). Adapters sit on the language tower, the vision encoder and the projector, so repair reaches whichever tower was cut. Selection uses 64 calibration images and the probe described above. TTFT is the median wall-clock time of a one-token generation over 30 calls after warm-up, on one fixed 896^2 image and one fixed question, batch 1, idle GPU. Repeated measurement varies by at most 1.6 ms. The three schedules of Appendix H all spend the same 768 steps. The *iterative* one repairs 96 steps after each of the 8 removals, and the *single pass* and our *accumulated* rule remove all 8 layers and then repair once.

The grounding column. PointArena and Where2Place both score whether a predicted coordinate lands where it should, on different scales, so their mean is a composite. We keep both so that the column does not rest on a single scorer.

Scorer corrections, applied to every criterion. Our scorer fixes 3 parsing faults of lmms-eval, identically for every criterion, and matches lmms-eval exactly on every item no fault touches. (i) Answers in normalised 0–1 coordinates are rescaled by the image size instead of being truncated to zero. (ii) The Where2Place scorer reads points written both as `(x, y)` and as `[x, y]`. (iii) A trailing period is stripped from ChartQA and InfoVQA predictions before the string match. The exact rules and the scorer are part of the code we intend to release (Reproducibility statement).

L.2 12 AND 16 REMOVED LAYERS

Table 3 removes 8 layers. Table 25 repeats the comparison at 12 and 16 layers, with the repair scaled to the budget (Appendix L.1) and everything else fixed. Every retention fraction below is taken against the unpruned model.

What changes with the budget. From 8 to 12 removals, our lead over the four baselines other than FinerCut widens on multiple choice on 2B and opens on 8B. It also widens on open-ended generation on 8B. Text reading remains the column we trail on 2B, each time to deepest-vision removal or angular distance. At 12, each of them is behind us or level on the other columns. The budget is hard on all of them: block influence, which puts all 12 removals into the language tower on 2B, is near zero on grounding, and the deepest-language rule is near zero on text reading. At 16, the two Reverse baselines lose text reading almost entirely (Table 25, bottom).

One summary survives all 3 budgets. Table 24 extends the worst-column summary of §5 to the harder budgets, for every criterion. The strongest of the four baselines other than FinerCut changes

with the budget. Deepest-vision removal, the strongest at 8, loses most of its advantage by 12. FinerCut is the only criterion that stays close to ours. It is ahead on 2B at 16 layers and, by a tenth, on 8B at 12.

	8 layers		12 layers		16 layers	
Criterion	2B	8B	2B	8B	2B	8B
ShortGPT	34.4	60.1	17.0	49.6	29.0	42.3
Angular	21.5	65.6	22.3	56.7	19.8	36.3
Reverse (ViT)	58.7	65.8	26.1	44.3	13.5	14.1
Reverse (LM)	34.0	37.1	13.0	33.0	11.0	18.8
FinerCut	61.8	68.8	52.4	**62.3**	**46.2**	52.5
Ours	**65.3**	**68.9**	**52.9**	62.2	38.1	**53.1**

Table 24: Each method's worst column, as a share of the unpruned model (%), at 3 budgets. A method that gives up one axis scores low on this summary. Methods and columns as in Table 3. Rows at 8 and 12 layers are three-seed means, and rows at 16 are one seed. Bold: best in the column.

Where the budget goes, and where the probe stops helping. Our split changes with the budget instead of scaling the 8-layer one. It takes a larger share from the language tower as the budget grows (the *Pruned* columns of Table 25). FinerCut, the same d_0 signal without our candidate step and probe, lands within two layers of ours at each budget, and at 16 it picks exactly our 8B split. The pattern is much the same at every budget. FinerCut wins text reading, and open-ended generation at 16 layers and on 8B at 12. Ours stays ahead of it on grounding at every budget, and on multiple choice everywhere except 8B at 8 layers. At 16 on 2B the probe stops helping (Table 24), and on 8B the two are level.

M BEYOND THE MAIN BACKBONES

Table 26 gives one number per backbone and repair objective, what the probe adds to its default candidates. Across the 14 language-model cells from 7B to 32B it loses 3, none by more than 0.6 points. It gains up to 6.8 at 25% removal, and 6.1 on Qwen3-14B and 11.1 on Qwen3-32B at 50%. On the vision–language models, from 2B to 38B in two families, it gains on every backbone and objective, by 1.3 to 11.0 points. The subsections below give the full tables for the sizes and the family the body does not show.

M.1 OTHER SIZES

We run the body's protocol unchanged on three more language models and two more vision–language models, with one seed per cell (Tables 27 and 28). Each table keeps the two comparisons this section is about: ours against ShortGPT, its probe ablation, and against removing the last layers.

The probe corrects its candidates where they need it. On Qwen3-14B the probe replaces 3 of the 10 layers block influence proposes and raises the average by 2.7 points. The gain comes from BBH (59.0 against 43.1). On Qwen3-32B block influence already proposes a nearly contiguous block early in the network (layers 8 to 24 of 64). The probe keeps 12 of its 16 layers, and the two land within a point of each other. At 50% removal the probe replaces 10 of the 32 layers block influence proposes, and ours keeps 63.5 of the unpruned model's 79.4 against 52.4 for ShortGPT, ahead on every column. On Qwen3-14B at 50% the probe replaces 8 of the 20 proposed layers, and ours is again ahead of ShortGPT on every column. On both models the gain grows with the removal rate (Table 26).

The safe layers move with size. Within one family, removing the last layers trails block influence by 2.2 points on Qwen3-8B (Table 2) and by 5.1 on Qwen3-14B. On Qwen3-32B it keeps 27.7 of the unpruned model's 79.4, while block influence keeps 78.5 (22.5 against 52.4 at 50%).

Backbone	Criterion	Pruned ViT	Pruned LM	Cost TTFT (ms)	Capability benchmarks (%) MCQ_8	$Text_4$	$Long_2$	$Ground_2$
InternVL3.5-2B	Unpruned	0	0	114	64.3	75.6	68.0	24.7
	ShortGPT	0	12	91 (−23)	42.0	46.5	25.2	4.2
	Angular	4–5	7–8	91 (−23)	51.8	55.6	31.6	5.5
	Reverse (ViT)	12	0	90 (−24)	50.6	19.7	24.3	16.8
	Reverse (LM)	0	12	91 (−23)	33.2	9.8	11.7	5.2
	FinerCut	9	3	90 (−24)	50.8	**57.1**	35.6	18.1
	Ours	10	2	90 (−24)	**56.9**	54.6	**36.0**	**21.6**
InternVL3.5-8B	Unpruned	0	0	222	70.6	79.6	74.9	38.3
	ShortGPT	1–2	10–11	173 (−49)	51.3	60.5	41.7	19.0
	Angular	5	7	182 (−40)	60.7	67.8	43.2	21.7
	Reverse (ViT)	12	0	199 (−23)	58.2	39.7	33.2	26.6
	Reverse (LM)	0	12	169 (−53)	61.4	48.5	24.7	16.8
	FinerCut	7–8	4–5	189 (−33)	65.1	**70.8**	**46.7**	25.0
	Ours	8	4	188 (−34)	**66.8**	68.2	46.6	**27.8**

Backbone	Criterion	Pruned ViT	Pruned LM	Cost TTFT (ms)	Capability benchmarks (%) MCQ_8	$Text_4$	$Long_2$	$Ground_2$
InternVL3.5-2B	Unpruned	0	0	114	64.3	75.6	68.0	24.7
	ShortGPT	1	15	83 (−31)	33.7	36.9	19.7	8.1
	Angular	5	11	82 (−32)	36.3	42.9	21.2	4.9
	Reverse (ViT)	16	0	82 (−32)	47.3	10.2	21.7	13.9
	Reverse (LM)	0	16	83 (−31)	31.9	8.3	11.7	4.9
	FinerCut	11	5	82 (−32)	47.0	**46.3**	**31.4**	13.5
	Ours	13	3	81 (−33)	**49.1**	30.3	25.9	**16.7**
InternVL3.5-8B	Unpruned	0	0	222	70.6	79.6	74.9	38.3
	ShortGPT	4	12	160 (−62)	49.7	54.7	34.2	16.2
	Angular	5	11	163 (−59)	53.0	59.0	33.1	13.9
	Reverse (ViT)	16	0	190 (−32)	51.3	11.2	23.3	**23.3**
	Reverse (LM)	0	16	150 (−72)	36.8	17.7	17.6	7.2
	FinerCut	9	7	174 (−48)	61.0	**63.7**	**40.8**	20.1
	Ours	9	7	173 (−49)	**62.2**	61.8	39.8	22.3

Table 25: Table 3 at harder budgets. **Top**: 12 removed layers (23.1% of 2B's blocks, 20.0% of 8B's), repair 1,152 steps, means of three seeds. **Bottom**: 16 removed (30.8% and 26.7%), repair 1,536 steps, one seed. At 12, ours leads multiple choice and grounding, leads long-form generation on 2B, and trails text reading. At 16 it still leads multiple choice on both backbones, but it gives up text reading and long-form generation, and grounding on 8B. Columns, scoring and corrections as in Table 3.

Below 2B the probe has little to measure. On Qwen1.5-1.8B-Chat ours lands within about a point of ShortGPT on every column. One plausible reason is that the probe needs a model that learns well from demonstrations. The ability to follow a rule from demonstrations that overrides a model's prior grows with scale (Wei et al., 2023b), and our probe is built from such rules. The comparison also has little room here. Two of the 3 pruned models beat the unpruned model on instruction following, so on that axis the repair data adds more than the removal costs, and the model's 24 layers leave the probe a pool of only 12.

The larger vision–language models repeat the body's split. On InternVL3.5-14B, block influence alone puts all 8 removals in the language tower, and the probe moves 6 of them to the vision tower. The probe gains 3.4 to 10.9 points on every column, as at 2B and 8B (§5). On InternVL3.5-38B, block influence again puts all 8 removals in the language tower, and the probe moves all 8 to the vision tower. The gain is smaller there, and ours is best on 3 of the 4 columns (Table 28). Eight layers are only about 7% of this network, against 12–15% for the 2B to 14B models, and ShortGPT already keeps most of the unpruned average. At both sizes, removing the last language layers costs the most, as it does on the smaller ones.

	Backbone	Seeds	Fine-tuning	Distillation	From
Language	Qwen1.5-1.8B-Chat	1	−0.5	–	Table 27
	Qwen3-8B	3	+6.3	+6.8	Table 2
	Qwen2-7B	3	+3.1	+2.2	Table 2
	Qwen1.5-7B	3	+2.2	+4.2	Table 2
	Llama-3.1-8B	3	+0.2	−0.6	Table 2
	Mistral-7B	3	−0.6	+1.6	Table 2
	Qwen3-14B	1	+2.7	–	Table 27
	Qwen3-14B (50%)	1	+6.1	–	Table 27
	Qwen3-32B	1	−0.5	–	Table 27
	Qwen3-32B (50%)	1	+11.1	–	Table 27
Vision–language	InternVL3.5-2B	3	+10.0	+11.0	Table 3
	InternVL3.5-8B	3	+8.1	+9.3	Table 3
	InternVL3.5-14B	1	+6.9	–	Table 28
	InternVL3.5-38B	1	+1.3	–	Table 28
	Qwen2-VL-2B	1	+4.4	–	Table 29
	Qwen2-VL-7B	1	+2.8	–	Table 29

Table 26: What the probe adds on each backbone: the Avg of ours minus the Avg of ShortGPT, which is ours without the probe, in points, under supervised fine-tuning and on-policy distillation. Each entry is the difference of two printed Avg values in the table named in the last column. Language models lose 25% of their layers (the rows marked 50% lose half), vision–language models eight. With three seeds, both Avg values are seed means.

Backbone	Criterion	MCQ_{12}	IFEval	GSM8K	BBH	Avg
Qwen1.5-1.8B-Chat	Unpruned	51.4	29.6	30.2	30.9	35.5
(24 → 18)	ShortGPT	**46.5**	**31.1**	**10.8**	**22.3**	**27.7**
	Reverse order	45.1	26.3	2.9	13.9	22.1
	Ours	45.4	**31.1**	10.0	22.2	27.2
Qwen3-14B	Unpruned	54.7	90.3	84.7	79.0	77.2
(40 → 30)	ShortGPT	**60.9**	61.9	**40.3**	43.1	51.5
	Reverse order	58.6	42.4	35.6	49.0	46.4
	Ours	57.8	**62.7**	37.3	**59.0**	**54.2**
Qwen3-14B	Unpruned	54.7	90.3	84.7	79.0	77.2
(40 → 20)	ShortGPT	47.9	40.4	12.1	17.5	29.5
	Reverse order	40.8	21.2	1.4	8.6	18.0
	Ours	**49.5**	**44.0**	**18.0**	**31.0**	**35.6**
Qwen3-32B	Unpruned	59.4	89.2	84.2	84.7	79.4
(64 → 48)	ShortGPT	**68.6**	**80.3**	**86.8**	**78.5**	**78.5**
	Reverse order	49.4	30.6	12.5	18.2	27.7
	Ours	68.0	79.9	86.7	77.4	78.0
Qwen3-32B	Unpruned	59.4	89.2	84.2	84.7	79.4
(64 → 32)	ShortGPT	57.6	58.0	53.1	41.0	52.4
	Reverse order	43.7	24.5	5.2	16.5	22.5
	Ours	**58.0**	**73.4**	**62.6**	**59.8**	**63.5**

Table 27: Other language-model sizes at 25% removal, and Qwen3-14B and Qwen3-32B also at 50% (40 → 20 and 64 → 32 layers), under the body's protocol (200 optimiser steps per removed layer in one repair), accuracy (%), one seed per cell. Columns as in Table 2; Avg is the mean of the 4 printed values. Reverse order removes the last layers. Bold: best of the three criteria in each block and column. The unpruned row is not a criterion and is never bold.

Backbone	Criterion	Pruned ViT	Pruned LM	Cost TTFT (ms)	Capability benchmarks (%) MCQ_8	$Text_4$	$Long_2$	$Ground_2$	Avg
InternVL3.5-14B	Unpruned	0	0	329	72.2	79.8	71.9	43.1	66.8
	ShortGPT	0	8	276 (−53)	58.7	69.5	47.5	32.3	52.0
	Reverse (ViT)	8	0	311 (−18)	69.6	**76.2**	52.1	**37.1**	58.8
	Reverse (LM)	0	8	276 (−53)	**70.4**	58.9	42.6	22.9	48.7
	Ours	6	2	304 (−25)	69.6	74.9	**55.3**	35.7	**58.9**
InternVL3.5-38B	Unpruned	0	0	1088	75.6	83.2	77.5	44.3	70.2
	ShortGPT	0	8	1011 (−77)	74.3	**82.4**	65.8	40.8	65.8
	Reverse (ViT)	8	0	1005 (−83)	73.0	79.4	55.2	39.7	61.8
	Reverse (LM)	0	8	1012 (−76)	73.1	64.1	41.4	30.8	52.4
	Ours	8	0	1011 (−77)	**75.3**	81.8	**68.8**	**42.7**	**67.1**

Table 28: InternVL3.5-14B and InternVL3.5-38B, each with 8 layers removed and one 768-step repair per criterion, one seed, accuracy (%). Columns as in Table 3; Avg is the mean of the 4 printed values. Each Reverse row removes the last layers of one tower. InternVL3.5-38B's TTFT is measured with the model split over two GPUs. Bold: best of the four criteria in each block and column. The unpruned row is never bold.

M.2 A SECOND FAMILY, READ AGAINST THE FIRST

Table 29 puts the body's two InternVL3.5 sizes and both sizes of a second family, Qwen2-VL (Wang et al., 2024), into one grid with the same budget, repair, columns and scorer corrections. One held-out backbone can only say whether a rule transfers. Four can say how much a criterion's standing depends on the family it is measured in.

No baseline keeps its ranking when the family changes. Over the 16 backbone–axis cells, the rank vectors of the six criteria agree within a family and largely stop agreeing across families (mean Spearman 0.75 within, 0.19 across). Angular distance moves from near the bottom on InternVL3.5 to near the top on Qwen2-VL, while deepest-vision removal moves the opposite way. Ours is the criterion whose standing moves least. It is first in 7 of the 16 cells, in the top 3 in 15, never last, and closest to the cell winner on average and at worst.

What we lose on Qwen2-VL, and to whom. Angular distance wins multiple choice on both sizes and grounding on 7B. FinerCut wins text reading on both sizes. We win grounding on 2B and long-form generation on 7B. Against the seed spreads in the caption, the 7B multiple-choice and grounding gaps are within the spread, and the text-reading gaps are not. The selections explain the text-reading gap. On 2B we take all 8 layers from the vision tower. On this family, with a 32-layer vision tower over a 28-layer language tower, the vision-heavy cut costs more text reading than on InternVL3.5.

The probe's contribution transfers even where the lead does not. Against the ShortGPT row, the probe wins 7 of the 8 Qwen2-VL cells. The only loss is text reading on 2B, where block influence alone keeps the vision tower that text reading needs.

Reading the Qwen2-VL rows. First, the unpruned Qwen2-VL rows are not a usable reference on the generation axis. On a large share of MMVet items this family stops generating after the opening sentence, and the repair data almost cures this. Comparisons among repaired models are therefore sound, but unpruned-to-pruned comparisons are not. Second, Qwen2-VL-2B scores zero on Where2Place in every row, the unpruned model included, because it answers with the image corners. Its grounding column is therefore half its PointArena score.

N FULL PER-SET AND PER-CELL TABLES

This section holds the per-set scores behind the grouped columns of Tables 2 and 3 (Tables 30–32) and the per-axis schedule comparison behind Table 17 (Tables 33 and 34). The per-set tables use the

Backbone	Criterion	Pruned ViT	Pruned LM	Capability benchmarks (%) MCQ_8	$Text_4$	$Long_2$	$Ground_2$	Avg
InternVL3.5-2B	Unpruned	0	0	64.3	75.6	68.0	24.7	58.1
	ShortGPT	0	8	49.1	57.2	33.3	8.5	37.0
	Angular	1–2	6–7	54.6	60.6	32.7	5.3	38.3
	Reverse (ViT)	8	0	57.6	**66.1**	39.9	21.1	46.2
	Reverse (LM)	0	8	57.1	46.2	25.3	8.4	34.3
	FinerCut	6	2	55.4	**66.1**	42.0	19.4	45.7
	Ours	8	0	**58.2**	62.1	**44.4**	**23.2**	**47.0**
InternVL3.5-8B	Unpruned	0	0	70.6	79.6	74.9	38.3	65.8
	ShortGPT	1	7	56.4	66.9	47.4	23.0	48.4
	Angular	5	3	64.7	73.7	49.1	25.2	53.2
	Reverse (ViT)	8	0	67.0	75.1	49.3	32.0	55.8
	Reverse (LM)	0	8	68.1	60.4	39.4	14.2	45.5
	FinerCut	6–7	1–2	**68.2**	**75.6**	51.5	30.7	**56.5**
	Ours	7	1	67.6	74.0	**51.6**	**32.7**	**56.5**
Qwen2-VL-2B	Unpruned	0	0	56.3	72.4	45.1	25.8	49.9
	ShortGPT	1	7	46.9	57.4	33.5	9.6	36.9
	Angular	6	2	**55.4**	55.4	38.7	20.6	**42.5**
	Reverse (ViT)	8	0	46.7	12.1	27.6	10.0	24.1
	Reverse (LM)	0	8	50.9	41.7	18.0	6.9	29.4
	FinerCut	3	5	44.8	**58.1**	**40.2**	8.4	37.9
	Ours	8	0	53.3	52.0	38.3	**21.6**	41.3
Qwen2-VL-7B	Unpruned	0	0	65.6	79.8	56.1	33.3	58.7
	ShortGPT	4	4	57.3	61.9	47.1	20.5	46.7
	Angular	8	0	**61.8**	63.6	46.0	**26.7**	**49.5**
	Reverse (ViT)	8	0	51.9	14.3	29.1	14.6	27.5
	Reverse (LM)	0	8	52.5	16.7	15.5	6.8	22.9
	FinerCut	4	4	55.3	**68.0**	45.8	19.2	47.1
	Ours	7	1	61.5	62.4	**49.3**	24.9	**49.5**

Table 29: Eight layers removed, one 768-step repair per criterion, four backbones, accuracy (%), columns as in Table 3. InternVL3.5 rows are means of three seeds. Qwen2-VL rows are one seed, read against the seed spreads of the three-seed rows (median under 3 points per column). Grounding on Qwen2-VL uses that family's coordinate format. Bold: best of six; ties at the printed precision all bold. On the held-out family, angular distance and FinerCut take most columns. Ours collapses on no column and is second on Avg at 2B and tied first at 7B.

body's criteria, repair and three seeds at 25% removal (eight layers for the vision–language models). They repeat the grouped means in italics for reference only.

O PRIOR SELECTION SIGNALS, METHOD BY METHOD

Table 1 groups prior selection signals by what they predict. All of them except DarwinLM, which trains each candidate, are functionals of $f_{-S}(\cdot \mid \varnothing; W)$. Observation 1 shows that such a signal cannot order candidates by their effect on $D \mapsto \Delta W(D)$. FinerCut's output divergence (Zhang et al., 2024) is the d_0 of Eq. 2 measured without demonstrations. It is therefore both a baseline and the comparison that isolates the probe.

Present-damage signals. LaCo (Yang et al., 2024b) merges layers and LLM-Streamline (Chen et al., 2025c) replaces a pruned span with a trained module. Both change the repaired architecture, so they can be compared only end to end. BlockPruner (Zhong et al., 2025) removes attention and MLP sub-blocks by perplexity, a present-damage signal at a finer unit. CLP (Lu et al., 2025) learns a gate that picks one contiguous segment, using the output without exemplars, so it measures the same damage. Recent work refines the present-damage signal itself. Hinostroza et al. (2026) show that the cosine similarity behind block influence can misrank layers badly and score each layer by the accuracy lost when it is removed, and Ding et al. (2026) estimate each layer's Shapley contribution

Backbone	Criterion	MMLU	PIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	SciQ	RACE	BoolQ	TQA	MuSR	*MCQ_{12}*
Qwen3-8B	Unpruned	49.5	73.1	58.6	62.4	55.6	42.0	40.8	69.6	40.3	85.7	53.3	40.9	*56.0*
	ShortGPT	65.4	69.9	55.9	**63.3**	57.6	38.9	37.7	89.7	**41.5**	77.3	47.9	37.6	*56.9*
	Angular	**69.8**	69.0	54.8	62.4	58.6	39.0	34.6	90.6	41.0	82.4	**48.6**	42.6	*57.8*
	Reverse order	68.8	67.2	53.2	62.2	56.9	37.8	34.3	88.1	38.6	67.1	48.1	**44.4**	*55.6*
	FinerCut	50.8	**72.8**	**59.7**	60.3	**71.1**	**45.3**	**39.4**	**93.6**	39.0	76.8	46.0	34.6	*57.4*
	Ours	69.6	69.4	56.0	62.7	57.8	39.4	36.3	90.8	40.1	**86.7**	48.0	43.3	*58.3*
Qwen2-7B	Unpruned	69.0	77.6	75.3	65.4	61.3	48.0	46.0	68.3	45.6	86.4	56.1	39.0	*61.5*
	ShortGPT	34.9	74.6	59.1	56.7	53.8	37.9	39.0	76.6	**36.8**	59.2	43.4	**44.2**	*51.3*
	Angular	31.5	75.2	59.8	55.8	**58.3**	38.7	**41.4**	**82.2**	35.2	46.6	42.2	43.8	*50.9*
	Reverse order	36.4	64.0	44.1	56.8	38.9	29.0	34.4	55.1	33.3	**72.2**	**45.2**	34.6	*45.3*
	FinerCut	**37.1**	**75.9**	60.3	58.0	53.8	**39.4**	40.4	77.4	34.9	56.4	42.5	37.3	*51.1*
	Ours	36.9	72.3	**61.0**	**61.7**	51.2	38.3	38.1	75.7	36.0	63.4	44.0	38.6	*51.4*
Qwen1.5-7B	Unpruned	60.4	71.2	63.8	57.1	51.8	40.4	43.6	63.2	43.7	85.2	57.0	38.0	*56.3*
	ShortGPT	56.1	70.4	57.2	63.6	47.2	35.9	39.7	70.5	41.1	**79.2**	45.0	34.4	*53.4*
	Angular	25.5	70.0	49.6	54.4	46.3	32.5	35.1	74.5	33.1	61.0	41.6	36.0	*46.6*
	Reverse order	**56.4**	69.0	54.7	63.6	44.2	33.4	36.9	73.0	38.9	74.4	**49.8**	**37.8**	*52.7*
	FinerCut	44.9	**72.2**	**59.3**	63.0	**49.7**	**37.7**	**40.0**	**77.9**	**44.0**	70.2	45.3	34.7	*53.2*
	Ours	52.6	71.7	58.8	**64.3**	48.1	37.3	**40.0**	73.8	42.6	77.8	46.4	36.3	*54.1*
Llama-3.1-8B	Unpruned	63.2	79.4	72.5	67.2	75.9	54.0	49.0	91.9	46.3	83.7	55.1	38.8	*64.8*
	ShortGPT	**64.7**	73.2	63.6	66.7	62.8	44.1	39.1	86.1	**45.0**	86.5	46.2	41.2	*59.9*
	Angular	64.0	73.2	**64.6**	**67.1**	**64.6**	**44.2**	**41.0**	86.8	44.9	**87.0**	46.9	**46.3**	*60.9*
	Reverse order	63.7	72.7	61.4	66.1	61.7	42.0	35.3	**88.3**	44.4	84.4	**49.7**	43.4	*59.4*
	FinerCut	59.9	**75.1**	64.3	65.6	61.8	43.3	39.2	86.5	43.9	83.0	44.3	41.8	*59.1*
	Ours	64.0	73.6	62.7	66.9	61.8	43.5	38.5	84.4	**45.0**	85.7	46.7	41.9	*59.6*
Mistral-7B	Unpruned	59.9	77.8	74.8	66.7	71.3	53.1	43.4	85.9	47.9	86.7	66.4	36.9	*64.2*
	ShortGPT	**55.5**	73.5	64.2	68.0	52.4	39.2	36.7	81.5	**41.6**	69.6	**51.2**	40.9	*56.2*
	Angular	**55.5**	73.5	64.2	67.9	52.4	39.2	36.7	81.7	41.5	69.6	**51.2**	40.9	*56.2*
	Reverse order	53.6	73.0	62.3	66.2	53.8	38.7	37.0	**84.4**	38.0	**79.0**	50.6	**43.9**	*56.7*
	FinerCut	46.4	**74.6**	**65.1**	64.6	**54.5**	**40.8**	37.3	81.6	38.3	69.7	48.3	37.3	*54.9*
	Ours	54.7	72.7	64.4	**68.8**	52.4	39.7	**37.8**	82.3	41.1	77.6	49.8	41.7	*56.9*

Table 30: The 12 multiple-choice sets behind MCQ_{12} (Table 2), one column each, with MCQ_{12} repeated in italics for reference. Bold: best criterion in the column within a backbone. Several unpruned cells are below the pruned models because the unpruned chat model does not produce the letter format these sets score, and the repair data teaches it. MCQ_{12} is the plain mean of the 12 columns.

Backbone	Criterion	MMStar	BLINK	SciQA	MMVP	AI2D	Hallu.	RWQA	MMMU	*MCQ_8*
InternVL3.5-2B	Unpruned	59.1	50.5	91.3	74.0	76.2	54.7	60.7	48.2	*64.3*
	ShortGPT	41.2	39.4	62.0	61.9	55.1	48.6	51.1	33.7	*49.1*
	Angular	48.6	41.0	73.8	66.0	64.0	54.7	52.4	36.7	*54.6*
	Reverse (ViT)	46.0	47.0	81.0	64.3	68.5	55.6	53.6	44.5	*<u>57.6</u>*
	Reverse (LM)	47.9	45.8	81.8	65.8	66.3	51.5	56.0	41.4	*57.1*
	FinerCut	45.1	42.6	77.5	64.6	66.2	52.1	54.2	40.7	*55.4*
	Ours	46.1	46.8	82.7	64.3	69.9	55.8	54.8	44.7	***58.2***
InternVL3.5-8B	Unpruned	66.1	57.9	97.2	79.0	82.4	60.7	64.7	56.7	*70.6*
	ShortGPT	51.7	44.9	68.5	69.1	60.5	57.7	61.7	37.1	*56.4*
	Angular	57.0	51.7	88.7	72.6	75.4	60.4	60.3	51.9	*64.7*
	Reverse (ViT)	57.4	55.3	94.7	73.7	78.6	59.4	62.0	55.0	*67.0*
	Reverse (LM)	61.1	54.4	96.6	76.9	81.2	56.7	63.2	54.9	*<u>68.1</u>*
	FinerCut	60.1	55.4	95.5	74.4	79.9	61.0	64.0	55.4	***68.2***
	Ours	58.7	55.9	95.4	73.8	79.4	59.5	62.7	55.0	*67.6*

Table 31: The 8 letter-choice sets behind MCQ_8 (Table 3), 8 layers removed, three seeds. The last column is their mean, the axis the body reports. Bold and underline: best and second best of the six criteria on that mean.

to perplexity with a surrogate trained on sampled removal sets. Both score the unrepaired model, so neither asks which removal repair will undo.

Signals that estimate recovery. Two lines of work estimate recovery. PRACTISE (Wang & Wu, 2024) fits adaptors around each dropped CNN block, and DarwinLM (Tang et al., 2026), a width-pruning method, gives each candidate a brief training run. Appendix G.1 runs a depth port of

		Text$_4$					Long$_2$			Ground$_2$		
Backbone	Criterion	OCRBench	ChartQA	TextVQA	InfoVQA	*Text$_4$*	MMVet	LLaVA-W	*Long$_2$*	PointArena	Where2Place	*Ground$_2$*
InternVL3.5-2B	Unpruned	82.1	82.5	74.5	63.5	*75.6*	60.4	75.7	*68.0*	45.7	3.8	*24.7*
	ShortGPT	62.0	67.3	56.4	43.3	*57.2*	26.5	40.2	*33.3*	16.5	0.4	*8.5*
	Angular	65.4	69.9	59.2	48.2	*60.6*	27.6	37.8	*32.7*	10.6	0.0	*5.3*
	Reverse (ViT)	71.6	76.2	63.5	53.1	***66.1***	33.5	46.2	*39.9*	38.3	3.9	*21.1*
	Reverse (LM)	42.3	61.7	38.5	42.3	*46.2*	21.7	28.9	*25.3*	16.7	0.0	*8.4*
	FinerCut	73.7	74.4	65.4	50.8	*66.1*	35.9	48.1	*42.0*	37.1	1.8	*19.4*
	Ours	63.4	75.5	58.5	50.8	*62.1*	37.3	51.6	***44.4***	41.5	4.8	***23.2***
InternVL3.5-8B	Unpruned	82.7	86.7	76.6	72.4	*79.6*	66.9	82.8	*74.9*	55.3	21.3	*38.3*
	ShortGPT	71.5	76.5	64.6	55.0	*66.9*	35.2	59.6	*47.4*	41.4	4.6	*23.0*
	Angular	79.1	81.5	69.9	64.2	*73.7*	40.7	57.5	*49.1*	41.6	8.8	*25.2*
	Reverse (ViT)	79.4	84.5	70.1	66.2	*75.1*	42.7	56.0	*49.3*	48.7	15.3	*32.0*
	Reverse (LM)	57.3	73.4	49.6	61.3	*60.4*	34.3	44.5	*39.4*	26.9	1.5	*14.2*
	FinerCut	79.8	84.1	70.4	68.0	***75.6***	45.9	57.0	*51.5*	50.3	11.0	*30.7*
	Ours	76.3	84.1	68.7	67.0	*74.0*	45.0	58.2	***51.6***	49.2	16.3	***32.7***

Table 32: The other three axes of Table 3 set by set, 8 layers removed, three seeds: text reading, open-ended generation and grounding, each followed by its mean. Bold and underline as in Table 31. The two tables together reproduce every axis of Table 3.

Pool and schedule	MCQ$_{12}$	IFEval	GSM8K	BBH
Block-influence pool, single pass (default)	**56.1**	**57.0**	41.3	**48.8**
accumulated	55.4	56.9	39.5	42.2
Angular-distance pool, single pass	55.8	55.5	**42.7**	48.0
accumulated	55.4	56.2	40.2	42.9

Table 33: **The one-stack comparison of Table 17 per axis.** Mean of the five language models, three seeds, 25% removal, identical total repair budget. Per-backbone means of the default pool, and the iterative schedule, are in Table 17. Rows apply Eq. 3 to a pool built from block influence (the default) or angular distance. Re-scoring block influence each round, as the accumulated rows do, gives the same pool. The d_0 pool is compared in Table 14. Bold: best of the four rows in the column.

Setting	Backbone	Selection schedule	ViT/LM	MCQ$_8$	Text$_4$	Long$_2$	Ground$_2$	Avg
8 layers, SFT	2B	accumulated (default)	8/0	58.2	62.1	**44.4**	**23.2**	**47.0**
		single pass, same pool	7/1	**58.3**	**66.2**	39.0	19.2	45.7
		iterative	7/1	58.1	59.5	40.2	22.9	45.2
	8B	accumulated (default)	7/1	67.6	74.0	**51.6**	32.7	**56.5**
		single pass, same pool	6–7/1–2	67.3	**74.4**	50.8	31.5	56.0
		iterative	6–7/1–2	**68.6**	73.7	48.7	**32.9**	56.0
12 layers, SFT	2B	accumulated (default)	10/2	**56.9**	**54.6**	**36.0**	**21.6**	**42.3**
		single pass, same pool	11/1	55.4	52.6	**36.0**	18.8	40.7
	8B	accumulated (default)	8/4	**66.8**	68.2	**46.6**	**27.8**	**52.4**
		single pass, same pool	8–9/3–4	64.9	**68.4**	44.7	25.4	50.9
16 layers, SFT	2B	accumulated (default)	13/3	49.1	**30.3**	25.9	**16.7**	**30.5**
		single pass, same pool	14/2	**51.2**	29.5	**26.7**	12.9	30.1
	8B	accumulated (default)	9/7	**62.2**	**61.8**	**39.8**	22.3	**46.5**
		single pass, same pool	11/5	61.6	52.1	37.6	**24.6**	44.0

Table 34: **The two-tower comparison of Table 17 at 3 budgets.** InternVL3.5 at 8, 12 and 16 removed layers under supervised repair: the accumulated rule of §5 (the default) against a single pass over the same pool and probe (T per tower, 192 probe items) and, at 8 layers, the iterative schedule, which repairs 96 steps after each removal. Removal and repair as in Table 3. Three seeds, one at 16 layers. *ViT/LM*: layers taken per tower (a range where seeds disagree). Bold: best in the block.

DarwinLM under our protocol. We ask the same question with a forward pass, the cheaper estimate when repair dominates the budget, as it does here.

Vision–language models. ECoFLaP (Sung et al., 2024) spreads an unstructured sparsity budget over both towers from a global score, UPop (Shi et al., 2023) learns per-modality pruning ratios for rows and columns, and Short-LVLM (Ma et al., 2025) removes language-tower layers chosen on vision–language tokens. None splits block removals between the towers, as §5 does.