RECAL: CALIBRATING STRUCTURED PRUNING FOR ON-POLICY DISTILLATION RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured pruning reduces the deployment cost of reasoning language models, but capability degradation can hinder subsequent on-policy distillation (OPD) recovery. Because OPD relies on student-generated trajectories, damage that persists after offline distillation can limit its effectiveness. We propose RECAL, Recovery-Aware Calibration, a simple plug-and-play approach that improves OPD recovery by adjusting calibration before pruning. RECAL uses forward KL between an unpruned teacher and a pruned probe to identify disrupted teacher-supported predictions, then reweights calibration statistics to guide existing pruning criteria toward preserving them. Across models and pruning methods, RECAL improves mathematical reasoning after OPD in most comparisons, with gains of up to 16.7 percentage points on AIME and improvements in most code-generation comparisons. Further analysis shows reduced residual damage at heavily affected tokens and performance advantages that persist through recovery. These results support calibrating structured pruning for OPD recovery. Our code: https://anonymous.4open.science/r/ReCal-hy.

1 INTRODUCTION

Large language models (LLMs) increasingly support complex reasoning and autonomous agent tasks, but their growing size and long generations make deployment expensive (Anthropic, 2026; OpenAI, 2026; Yang et al., 2025). Structured pruning reduces this cost by removing entire channels, heads, or layers, producing smaller models that run on standard hardware (Ma et al., 2023; An et al., 2024; Ashkboos et al., 2024). However, pruning often requires subsequent training to recover lost capabilities (Xia et al., 2024; Muralidharan et al., 2024; Sreenivas et al., 2024). Recent distillation advances support a two-stage recovery pipeline: offline distillation through supervised fine-tuning (SFT) on teacher responses, followed by on-policy distillation (OPD) on student-generated trajectories (Kim & Rush, 2016; Agarwal et al., 2024; Zhang et al., 2026). The unpruned model serves as teacher and the pruned model as student.

Despite this progress, pruning can severely degrade reasoning, and offline SFT only partly restores accuracy (Figure 1b). Pruned models can produce repetitive, unfinished, or incorrect responses (Wang et al., 2025; Zhang et al., 2026). Worse still, OPD obtains teacher supervision at these student-generated prefixes, where poor generations can compromise the feedback (Xu et al., 2025). A teacher can agree with the student's next-token predictions at a flawed prefix even when the reasoning fails; teacher-confirmed repetition is one documented example (Zhang et al., 2026). Low reverse KL on student rollouts therefore need not indicate successful reasoning. In our baseline evaluations, many responses still reach the length limit after OPD (Figure 1c), leaving substantial generation failures unresolved.

Since recovery depends on student trajectories, pruning should account for damage that may hinder recovery, complementing training-aware structure selection (Tang et al., 2026; Ai et al., 2026). We examine this damage on teacher-generated prefixes, where the teacher and a pruned model receive the same context. Forward KL emphasizes predictions that the teacher supports but the pruned model assigns little probability. Our analysis shows that tokens most affected by pruning remain a major source of discrepancy after recovery. Giving these tokens greater calibration weight could therefore protect vulnerable predictions before recovery begins.

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.

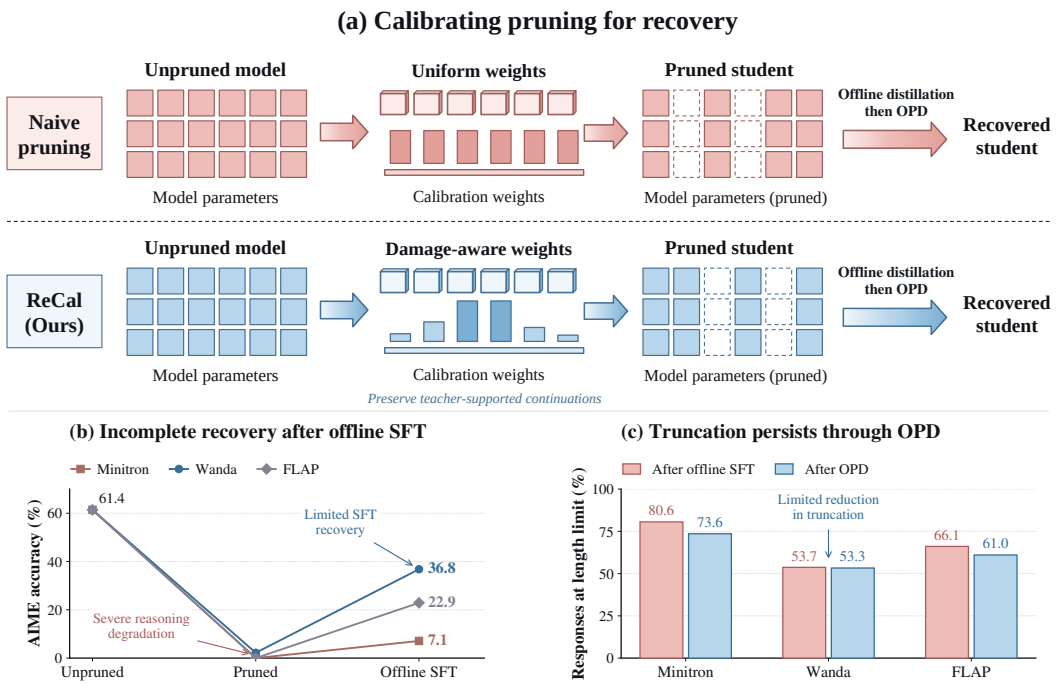


Figure 1: **Pruning and the limitations of subsequent recovery.** (a) RECAL adjusts calibration before SFT and OPD. (b) AIME accuracy before pruning, after pruning, and after offline SFT. (c) The fraction of responses reaching the length limit after SFT and OPD. Both data panels show measured 8B baselines; Appendix B.4 gives the protocol.

We therefore propose RECAL, a simple *Recovery-Aware Calibration* method for structured pruning. As shown in Figure 1(a), RECAL first uses a pruned probe to measure token-level forward KL on teacher trajectories. These scores reweight calibration statistics used to recompute channel importance on the unpruned model. Applying the new mask produces a student for the same SFT and OPD recovery pipeline. RECAL is plug-and-play: it preserves the base pruning criterion and target widths while changing the evidence used to select channels.

To evaluate RECAL, we compare its recovery performance with representative structured pruning baselines on Qwen3-8B and Qwen3-4B-Instruct. After on-policy distillation, RECAL achieves stronger mathematical reasoning than the best-performing baseline on both models. It also improves mathematical reasoning in seven of eight model–criterion comparisons, with gains of up to 16.7 percentage points, and benefits code generation in most comparisons. These results show that calibrating which structures survive pruning can improve the capabilities recovered by the subsequent distillation pipeline.

2 PRELIMINARIES

We first introduce calibration-based structured pruning in Section 2.1, then describe the on-policy distillation recovery process that follows it in Section 2.2.

2.1 STRUCTURED PRUNING

Structured pruning removes parameter groups to obtain a smaller model (Ma et al., 2023; An et al., 2024; Muralidharan et al., 2024). We consider feed-forward network (FFN) width pruning, where each selected channel is removed together with its associated input and output weights. Let p_T denote the original model and $m^\ell \in \{0,1\}^{d_\ell}$ the channel mask in layer ℓ. A value of one retains a channel. Applying these masks to the original weights gives the pruned model p_m.

Calibration-based criteria use a small set of inputs to estimate channel importance. For an additive criterion, let c_{ij}^{ℓ} be the contribution of calibration token i to channel j in layer ℓ. Aggregating over N tokens and retaining the k_ℓ highest-scoring channels gives

$$s_j^{\ell} = \frac{1}{N}\sum_{i=1}^{N} c_{ij}^{\ell}, \qquad m_j^{\ell} = \mathbb{1}\big[j \in \mathrm{Top}_{k_\ell}(s^{\ell})\big]. \tag{1}$$

The contribution can use activation magnitude, norms, fluctuations, or gradients (Sun et al., 2024; An et al., 2024; Ma et al., 2023). Equation 1 covers additive scores; see Appendix A.2 for other criteria.

We reweight token evidence while preserving the criterion and target widths.

2.2 On-Policy Distillation Recovery

Offline sequence distillation first initializes recovery by fitting fixed responses from the unpruned teacher (Kim & Rush, 2016). We implement this stage as supervised fine-tuning (SFT) on the teacher responses; we use SFT to denote this offline distillation stage throughout the paper. On-policy distillation (OPD) then trains the student on responses sampled from its own policy, with a frozen teacher providing the supervision (Agarwal et al., 2024; Li et al., 2026). Let p_θ be the trainable pruned student. For prompt x, a rollout checkpoint $p_{\bar{\theta}}$ generates response y, and both teacher and student evaluate the prefixes $h_t = (x, y_{<t})$. A full-vocabulary reverse-KL matching objective on these rollouts is

$$\mathcal{L}_{\mathrm{OPD}}(\theta;\bar{\theta}) = \mathbb{E}_{x,\, y\sim p_{\bar{\theta}}(\cdot|x)}\left[\frac{1}{|y|}\sum_{t}\mathrm{KL}\big(p_\theta(\cdot \mid h_t)\|p_T(\cdot \mid h_t)\big)\right], \tag{2}$$

where $\mathrm{KL}(p\|q) = \sum_v p(v)\log[p(v)/q(v)]$. Fresh responses are collected as the student updates, so the teacher supervises prefixes reached by the student's current policy.

The retained structure determines the starting student for this process. We hold recovery fixed and examine whether pruning damage can inform structure selection. Appendix A specifies the practical OPD signal and training settings.

3 Understanding Pruning Damage and Recovery

We analyze several structure pruning methods on held-out teacher responses to locate pruning damage and track its recovery. At token t in response r, both models receive the same prefix $h_{r,t}$. Token-level damage is

$$d_{r,t}(m) = \mathrm{KL}\big(p_T(\cdot \mid h_{r,t})\|p_m(\cdot \mid h_{r,t})\big). \tag{3}$$

This measures the change in next-token predictions, weighted by the teacher distribution. Within each response, we rank tokens by raw-pruned baseline KL and divide them into five equal-count groups, Q1–Q5, from lowest to highest damage. Each group contains one fifth of the response's tokens. We keep memberships and teacher prefixes fixed through offline SFT and OPD, averaging KL within each response/group and then across responses. Concentration is examined in both models; recovery tracking uses Wanda. Full settings are in Appendix A.6.

Pruning distorts some token predictions much more than others. Figure 2(a) shows that the highest-damage 20% of tokens contribute 81.5–82.9% of within-response damage across the three criteria. The second model exhibits the same pattern, with shares of 80.9–83.1% (Appendix A.6). Channel removal therefore affects token predictions unevenly, which uniform aggregation does not explicitly distinguish. We next ask whether these predictions remain damaged after recovery.

Recovery reduces token-level damage but leaves substantial residual errors. Using the fixed groups defined above, Figure 2(b) shows that mean KL in the highest-damage group decreases from 2.110 after pruning to 0.705 after SFT and 0.627 after OPD. The full pipeline removes 70.3% of this group's initial discrepancy, with most of that reduction occurring during SFT. Nevertheless, its post-OPD KL remains nearly three times that of the next-highest group (0.214). Substantial repair therefore leaves these tokens disproportionately damaged.

Tokens most affected by pruning still account for most damage after recovery. To quantify how much of the remaining damage can be located before recovery, we measure the share carried by the

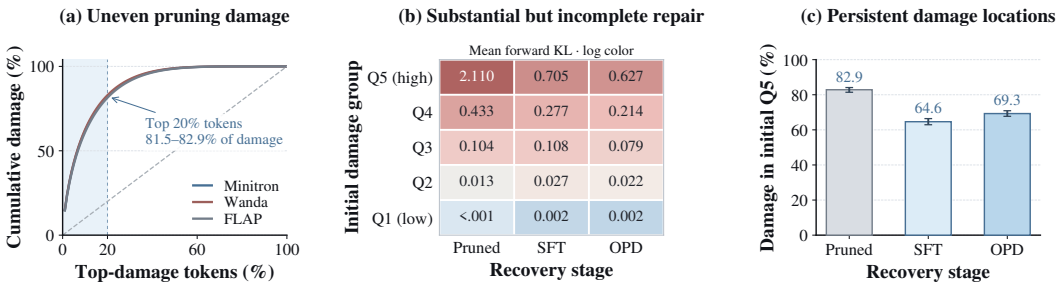


Figure 2: **Locating damage before recovery.** (a) Within-trajectory damage concentration across ordinary pruning criteria. (b) Mean KL through baseline recovery in groups fixed by initial damage; color is logarithmic. (c) The fraction of each response's damage carried by its initially highest-damage group (Q5), averaged across responses. Q1–Q5 are equal-count groups ordered by initial damage. Token groups are never reselected after recovery. SFT denotes offline distillation through supervised fine-tuning. The pale vertical band marks Q5; curve bands and error bars denote 95% trajectory-bootstrap intervals.

original highest-damage group. Without reselecting any tokens, this group accounts for 64.6% of within-response damage after SFT and 69.3% after OPD (Figure 2c). The increase in share from SFT to OPD does not mean its damage grows: its absolute KL decreases, while its share of the remaining discrepancy rises. Initial damage thus identifies tokens that remain poorly matched after recovery, using a signal already available before training.

These observations motivate accounting for teacher-prefix damage during structure selection. A base-pruned model can identify affected tokens; emphasizing their calibration evidence may reduce the discrepancy passed to recovery. Section 4 defines this intervention, and Section 5 tests whether it improves recovered accuracy.

4 Recovery-Aware Calibration for Pruning

RECAL adjusts calibration to address the persistent damage identified in Section 3. We first locate token-level damage, use it to calibrate structure selection, and then recover the selected student through SFT and OPD. Figure 3 summarizes this procedure.

Locate damage. We apply the base pruning criterion at the target widths to obtain a probe p_{m_0}. The teacher and probe evaluate identical prefixes from teacher-generated responses. We estimate the forward-KL damage in Equation 3 using teacher-selected tokens and a residual-probability bucket (Appendix A.2), and denote the resulting scores by $d_{r,t}$. Teacher trajectories expose damage independently of the probe's own rollouts. The probe does not restrict the final mask.

Calibrate & prune. We convert the probe's damage into calibration weights. For response r with T_r valid tokens, let $D_r = \sum_t d_{r,t}$. The token weights and resulting additive channel scores are

$$\widetilde{w}_{r,t} = \frac{d_{r,t}}{D_r}, \qquad \sum_{t=1}^{T_r} \widetilde{w}_{r,t} = 1. \tag{4}$$

$$s_j^\ell = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \widetilde{w}_{r,t} c_{r,t,j}^\ell, \qquad (m^\star)_j^\ell = \mathbb{1}\big[j \in \mathrm{Top}_{k_\ell}(s^\ell)\big]. \tag{5}$$

Here R is the number of calibration responses and $c_{r,t,j}^\ell$ is the base contribution defined in Section 2.1. Each response has equal total weight, with greater weight assigned to its more damaged tokens. For negligible total damage, we use uniform weights (Appendix A.2).

The weighted statistics are computed on the original unpruned model, and the selected mask is applied to its original weights. Thus, a channel removed by the probe remains eligible for the final student. Non-additive criteria incorporate the weights into their activation statistics or token losses before aggregation (Appendix A.2). The base criterion and retained widths remain unchanged. Appendix C connects teacher-prefix forward KL to sequence preservation and states its limits.

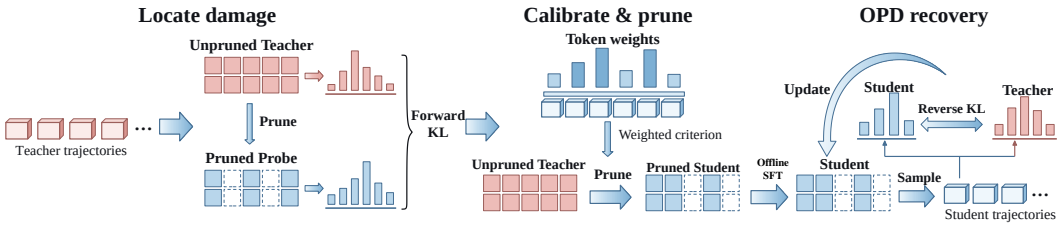


Figure 3: **Overview of ReCal.** *Locate damage:* compare the teacher and a pruned probe on teacher trajectories. *Calibrate & prune:* use token-level damage to reweight the base criterion and prune the original model. *OPD recovery:* recover the selected student with the unchanged training procedure.

OPD recovery. The selected student first undergoes offline distillation through SFT on fixed teacher responses and then OPD with the same unpruned teacher. Both stages use the same training procedure as the baseline. ReCal changes the structure supplied to recovery, while OPD continues to learn from student-generated trajectories as in Section 2.2. Appendix A gives training details.

5 Experiments

After introducing the setup in Section 5.1, we address four questions in Sections 5.2–5.5:

- **RQ1:** How does ReCal compare with existing pruning methods before and after recovery? Does it improve the performance reached after OPD?
- **RQ2:** Can ReCal improve different pruning criteria without changing their scoring rules? Do its benefits extend beyond the default Wanda configuration?
- **RQ3:** How does the benefit of ReCal change as more parameters are removed? Does it persist across compression ratios, models, and base criteria?
- **RQ4:** How does ReCal change the selected channels and the damage identified in Section 3? How do these changes relate to recovery?

5.1 Experimental Setup

Appendix A details the pruning adapters, data sources, training, and evaluation.

Models and baselines. We use Qwen3-8B in thinking mode and Qwen3-4B-Instruct-2507 without thinking (Yang et al., 2025). We compare structured FFN-channel adapters of Minitron, Wanda, FLAP, and LLM-Pruner (Muralidharan et al., 2024; Sun et al., 2024; An et al., 2024; Ma et al., 2023). All methods retain the same widths at a given compression ratio. The main comparison removes approximately 25% of total parameters; the ratio sweep also includes 15% and 35%.

Recovery and evaluation. Every student undergoes offline distillation through SFT on responses from its unpruned teacher, followed by OPD on student-generated trajectories. Paired methods share recovery data and training settings. We evaluate mathematics on AIME 2024–2026, science on GPQA-Diamond (Rein et al., 2023), instruction following on IFBench (Pyatkin et al., 2025), and code generation on LiveCodeBench-v5 (Jain et al., 2025). We report AIME accuracy averaged over eight samples per question and the three years, GPQA accuracy, IFBench instruction-level strict accuracy, and LiveCodeBench pass@1. Supplementary experiments and additional results appear in Appendix B.

5.2 Recovery Performance (RQ1)

We use Wanda as the base criterion for the default ReCal configuration. Table 1 compares the three stages: Pruned, SFT, and SFT followed by OPD. We have the following observations:

- **Obs. 1: ReCal leads in mathematics after recovery.** After OPD, ReCal reaches 48.5% AIME accuracy in 8B and 37.2% in 4B, exceeding the strongest baseline, Wanda, at 41.7% and 32.4%. It leads on each AIME year in both models. After SFT, it already reaches 45.8% and 34.2%, respectively, establishing an advantage that persists through OPD.

Table 1: **Recovery at approximately 25% parameter reduction.** RECAL denotes Wanda + RECAL. Pruned: no recovery; SFT: offline distillation; OPD: SFT followed by OPD. Scores are percentages; Avg. averages all six benchmarks. Each column's best/second-best pruned scores within a model are **bold**/underlined, excluding unpruned references.

Method	Stage	Math			Science	Instruction	Code	Avg.
		AIME24	AIME25	AIME26	GPQA	IFB-S	LCB-v5	
Qwen3-8B								
Unpruned	—	66.2	55.8	62.1	54.5	23.0	46.4	51.3
Minitron	Pruned	0.0	0.0	0.0	1.0	16.9	0.0	3.0
	SFT	9.2	3.3	8.7	30.3	18.0	15.7	14.2
	OPD	16.7	10.0	15.4	35.9	21.2	19.9	19.9
FLAP	Pruned	0.0	0.0	0.0	23.2	16.3	0.0	6.6
	SFT	27.5	20.8	20.4	33.3	19.2	15.1	22.7
	OPD	31.7	25.4	30.0	43.4	20.1	24.7	29.2
LLM-Pruner	Pruned	0.4	0.0	0.0	22.2	16.0	0.0	6.4
	SFT	11.7	6.2	5.8	35.9	19.5	12.7	15.3
	OPD	18.8	7.9	10.4	34.8	**22.4**	17.5	18.6
Wanda	Pruned	2.5	2.1	2.1	26.8	20.6	4.2	9.7
	SFT	41.7	34.6	34.2	**45.5**	21.5	25.3	33.8
	OPD	46.7	36.7	41.7	42.4	19.5	25.9	35.5
RECAL	Pruned	33.3	23.7	20.0	35.4	20.1	16.9	24.9
	SFT	51.7	39.6	46.3	43.4	21.5	26.5	38.2
	OPD	**52.9**	**42.9**	**49.6**	40.4	18.3	**28.3**	**38.7**
Qwen3-4B-Instruct								
Unpruned	—	63.3	45.8	54.2	60.6	34.3	31.9	48.4
Minitron	Pruned	0.0	0.4	0.0	30.3	26.7	1.2	9.8
	SFT	6.2	0.8	5.4	42.9	28.2	15.7	16.5
	OPD	6.2	2.1	6.2	35.4	30.2	21.7	17.0
FLAP	Pruned	4.2	0.8	0.8	31.8	27.9	4.2	11.6
	SFT	21.7	11.7	13.8	36.9	26.2	21.7	22.0
	OPD	27.9	17.5	19.2	38.9	**30.8**	22.3	26.1
LLM-Pruner	Pruned	0.0	0.0	0.0	26.8	25.0	0.6	8.7
	SFT	6.7	2.5	6.7	31.8	25.9	15.1	14.8
	OPD	8.7	0.8	8.7	35.9	27.9	15.1	16.2
Wanda	Pruned	13.3	5.4	6.7	40.4	29.7	5.4	16.8
	SFT	29.6	23.7	20.4	**47.0**	27.3	20.5	28.1
	OPD	35.8	32.5	28.7	44.4	29.7	24.7	32.6
RECAL	Pruned	20.4	10.4	10.8	37.9	30.2	15.1	20.8
	SFT	41.7	31.7	29.2	43.9	27.3	22.9	32.8
	OPD	**42.5**	**34.2**	**35.0**	**47.0**	28.8	**25.3**	**35.5**

- **Obs. 2: Calibration preserves reasoning capability before recovery.** Immediately after pruning, the 8B default retains 25.7% AIME accuracy, compared with 2.2% for Wanda. The 4B scores are 13.9% and 8.5%. The advantage is already present before any recovery training, consistent with calibration preserving more of the teacher's reasoning capability in the selected structure.

5.3 PLUG-AND-PLAY IMPROVEMENT (RQ2)

We next apply RECAL to each pruning criterion while keeping its channel-scoring rule, target widths, and recovery procedure fixed. Figure 4 compares each fully recovered baseline with its calibrated counterpart. Baselines use prompt calibration, whereas RECAL uses weighted teacher trajectories; the comparison evaluates this combined intervention. We can draw the following observations:

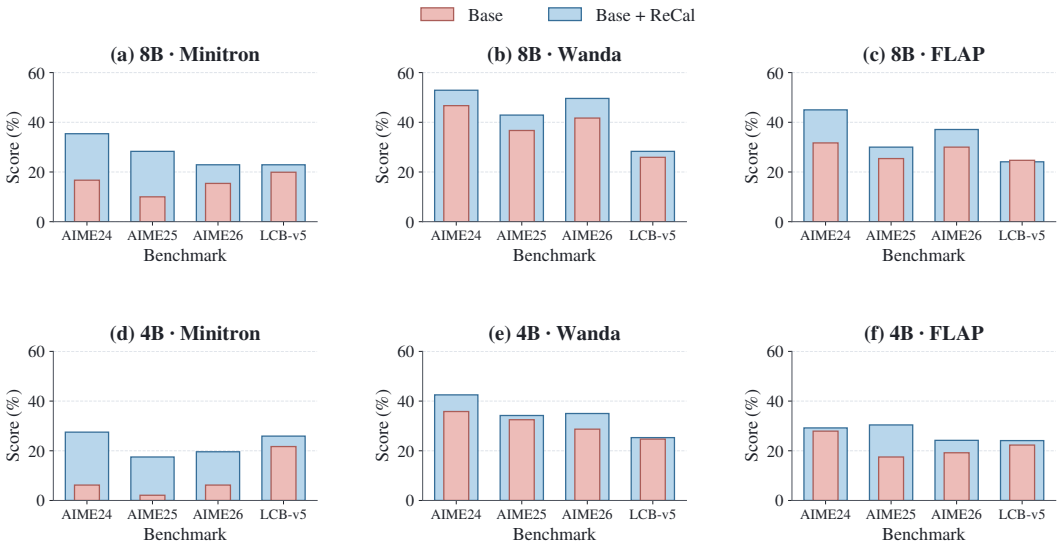


Figure 4: **Post-OPD performance across pruning criteria.** Measured results on AIME24–26 and LiveCodeBench-v5 (LCB-v5). Wide blue bars show Base + ReCal; narrow red bars show Base. Rows identify models and columns identify criteria. See Appendix B.1 for LLM-Pruner.

- **Obs. 3: ReCal improves mathematics in seven of eight pairs.** All six activation-based pairs improve on AIME (Figure 4), with gains up to 16.7 points. LLM-Pruner improves from 6.1% to 12.8% in 4B but declines from 12.4% to 9.2% in 8B (Appendix B.1). The benefit therefore extends across criteria, with a model-dependent exception.
- **Obs. 4: Code improves broadly, with trade-offs on other tasks.** LiveCodeBench improves in seven of eight pairs and GPQA in five, while IFBench declines in all eight (Figure 7c). For 8B LLM-Pruner, science and code gain 10.1 and 10.8 points despite the mathematics loss. These trade-offs limit claims of uniform capability preservation (Appendices B.1 and B.5).

5.4 Sensitivity to Compression Ratio (RQ3)

We vary whole-model parameter reduction from 15% to 35% for Minitron, Wanda, and FLAP in both models. Figure 5 shows both recovery stages. We can draw the following observations:

- **Obs. 5: ReCal improves SFT recovery at every measured ratio.** ReCal exceeds Base in all 18 model–criterion–ratio comparisons. Wanda gains 3.6, 9.0, and 14.6 points in 8B at 15%, 25%, and 35% reduction; 4B gains are 1.8, 9.6, and 7.6 points. Benefits persist across ratios but need not increase monotonically.
- **Obs. 6: More aggressive pruning still limits recovery.** All measured SFT curves decline with compression. At 35% reduction, 8B Wanda + ReCal retains 27.1% accuracy versus 12.5% for Base, while calibrated Minitron reaches only 3.6%. In 4B, calibrated Wanda also outperforms calibrated Minitron and FLAP at this ratio. Both the criterion and retained capacity still matter.

5.5 Structure and Recovery Analysis (RQ4)

We compare masks at identical widths (Figure 6) and revisit the teacher prefixes and baseline-defined Q1–Q5 groups from Section 3, now adding the calibrated checkpoints (Figure 7). We have the following observations:

- **Obs. 7: ReCal preserves most selections, with larger changes in deeper layers.** Activation-based pairs retain 75.8–79.3% of baseline deletions, with ranking correlations of 0.784–0.839. Replacement is greater in middle and late layers; 4B LLM-Pruner replaces 37.6% of its deletion set. With widths fixed, ReCal changes which channels survive while retaining most base selections.
- **Obs. 8: ReCal reduces the residual damage identified before recovery.** On the fixed Q5 tokens, ReCal lowers KL at every stage, ending at 0.470 versus 0.627 for Base, a 25.1% reduction. The

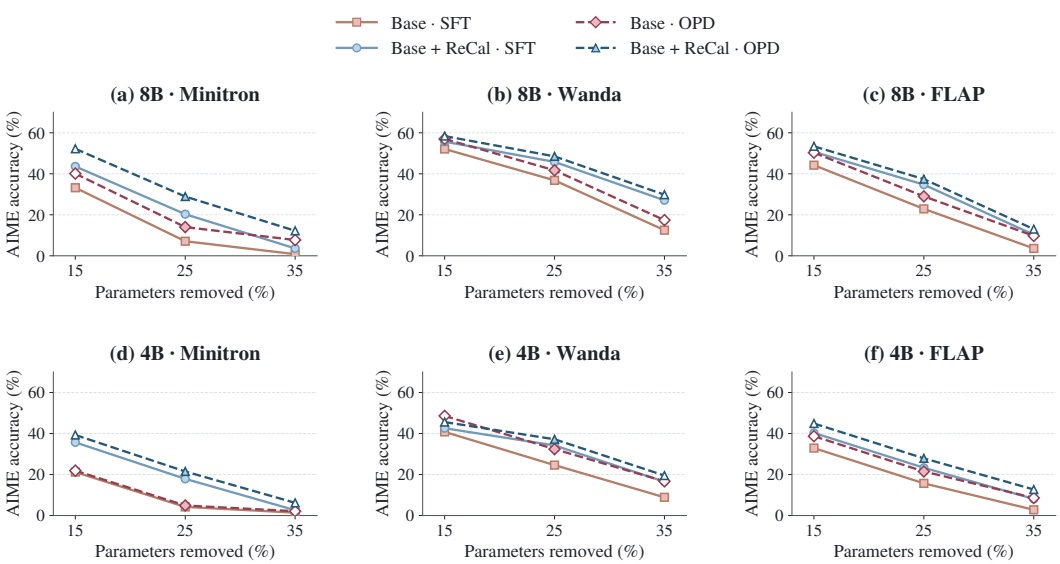


Figure 5: **Base versus Base + RECAL across compression ratios.** Solid: SFT; dashed: OPD. Ratios denote whole-model parameter reduction.

paired difference is -0.157 nats (95% interval: $[-0.177, -0.140]$). Q3–Q5 improve after OPD, while Q1/Q2 intervals include zero. The reduction thus reaches the vulnerable tokens identified in Section 3, without improving every group.

- **Obs. 9: The average SFT advantage persists after OPD.** Across all eight pairs, the mean AIME advantage is 7.9 points after SFT and 7.7 after OPD. The signed advantage increases in four pairs and decreases in four. In 8B LLM-Pruner, OPD narrows the SFT deficit without reversing it. Final gains therefore do not imply uniformly larger OPD increments. Appendix B.3 adds completion and alignment diagnostics.

6 RELATED WORK

Structured pruning and capability recovery. Structured pruning selects channels or heads using gradient, activation, or reconstruction criteria (Ma et al., 2023; An et al., 2024; van der Ouderaa et al., 2024; Meng et al., 2024; Ling et al., 2024; Guo et al., 2025). Continued training and distillation recover the resulting models (Xia et al., 2024; Muralidharan et al., 2024; Sreenivas et al., 2024). DarwinLM evaluates candidate structures after brief training, while NIRVANA models fine-tuning dynamics and selects calibration data with KL (Tang et al., 2026; Ai et al., 2026). Reasoning-oriented approaches use generated traces, preserve sampling diversity, or improve student initialization (Wang et al., 2025; Nguyen et al., 2025; Monjur et al., 2026; He et al., 2026). RECAL measures teacher–probe damage at individual tokens and reweights their contribution within each teacher trajectory. It preserves the base criterion and widths, changing the calibration evidence used to select a student for SFT and OPD. Appendix D compares these mechanisms and broader pruning approaches.

Offline and on-policy distillation. Offline distillation fits teacher responses (Kim & Rush, 2016); OPD supervises student-generated prefixes to address training–generation mismatch (Agarwal et al., 2024; Gu et al., 2024). DistiLLM uses skew divergences and adaptive reuse, DistiLLM-2 distinguishes teacher and student responses, and speculative distillation interleaves their tokens (Ko et al., 2024; 2025; Xu et al., 2025). Recent work studies thinking-pattern compatibility, entropy-aware forward KL, and privileged-context self-distillation (Li et al., 2026; Jin et al., 2026; Zhao et al., 2026). For pruned students, ShortOPD schedules short-to-long rollouts to reduce computation spent on teacher-confirmed repetition (Zhang et al., 2026). These methods modify recovery supervision or sampling. RECAL uses forward KL on teacher prefixes to select the starting structure, leaving the OPD loss unchanged.

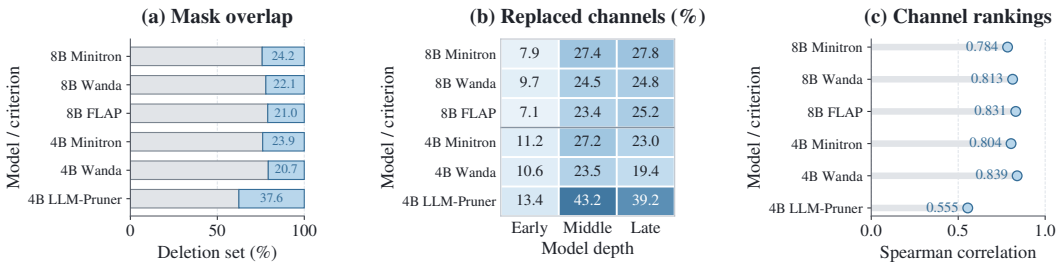


Figure 6: **Changes in channel selection.** (a) Shared (gray) and replaced (blue) baseline deletions. (b) Replacement by layer depth. (c) Spearman correlation of complete channel rankings.

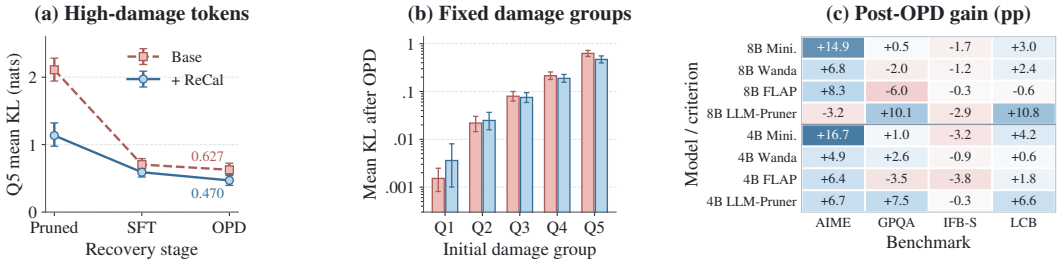


Figure 7: **Damage reduction and recovery.** (a) Q5 KL through recovery. (b) Post-OPD KL by fixed group (log scale). Both use 8B Wanda and Section 3 teacher prefixes; error bars are 95% trajectory-bootstrap intervals. (c) Post-OPD gains across eight pairs and four tasks: blue indicates gains and red indicates losses. AIME is the three-year mean.

7 Limitations

Generalization beyond the current pruning setting. Our experiments focus on FFN-width pruning for two Qwen3 models followed by a fixed SFT–OPD recovery pipeline. While ReCal consistently improves recovered reasoning performance across several pruning criteria, its effectiveness may depend on the type of structure being removed and the capability being evaluated. In particular, the mixed results on instruction following and the 8B LLM-Pruner case indicate that recovery-aware calibration does not uniformly benefit every criterion or task. An important next step is to extend ReCal beyond FFN channels to other structured compression settings, such as attention-head and layer pruning, and to study whether the same recovery-aware signal transfers across model families and more interactive agent workloads.

Understanding what makes recovery-aware calibration effective. ReCal combines teacher trajectories with damage-aware weighting based on forward KL, so the current experiments evaluate the complete calibration procedure rather than isolating each design choice independently. Although the fixed-prefix analysis shows that ReCal reduces residual discrepancy on tokens that are heavily affected by pruning, the precise connection between this reduction and downstream recovery remains only partially understood. We are therefore interested in disentangling the roles of trajectory choice, divergence direction, and token weighting, and in developing diagnostics that follow the student states encountered during OPD rather than only fixed teacher prefixes. These directions may help identify when recovery-aware calibration is most useful and lead to more general principles for selecting structures that remain recoverable after pruning.

8 Conclusion

We introduced ReCal, a recovery-aware calibration approach for structured pruning followed by on-policy distillation. By using teacher–probe disagreement to reweight calibration evidence before structure selection, ReCal improves recovered mathematical reasoning and reduces residual damage at vulnerable tokens. More broadly, our results suggest that pruning should account not only for what a model preserves immediately after compression, but also for what the resulting student can recover afterward.

AI USE STATEMENT

Generative AI tools were used solely as assistants for coding and writing. For coding, they supported the implementation, debugging, refactoring, and documentation of author-designed training and evaluation pipelines, data-processing utilities, and visualization scripts. For writing, they were used to improve clarity, concision, phrasing, and copy-editing of content and claims developed by the authors. Generative AI was not used for research ideation, problem formulation, methodological or experimental design, theoretical development or proofs, literature search or reference selection, result interpretation, or the generation of experimental results. All AI-assisted code was reviewed and tested by the authors, and all reported results were independently verified against the underlying per-sample records. The authors reviewed all AI-assisted outputs and take full responsibility for the final manuscript, claims, code, and associated artifacts.

ETHICS STATEMENT

This study examines compression and recovery of existing language models using public benchmarks and teacher-generated responses. Compression may reduce the resources needed to deploy capable models, including in applications with potential for misuse. The reported recovery gains do not establish safety, fairness, or reliability in deployment. In particular, instruction-following and scientific reasoning do not improve uniformly, and the compressed students retain failure modes of both pruning and their teachers. Models and data should be used under their applicable licenses, and intended applications require separate capability and safety evaluation.

REPRODUCIBILITY STATEMENT

Our code: https://anonymous.4open.science/r/ReCal-hy. Sections 4 and 5.1 describe the method and evaluation comparisons. Appendix A documents the models, calibration adapters, training data, recovery settings, and evaluation protocol; Appendix A.6 specifies the fixed-prefix analysis. The accompanying source bundle includes aggregate measurements and table/figure scripts. Appendix C states the theoretical assumptions and provides the proof.

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*, 2024. URL https://arxiv.org/abs/2306.13649.

Mengting Ai, Tianxin Wei, Sirui Chen, and Jingrui He. NIRVANA: Structured Pruning Reimagined for Large Language Model Compression. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2509.14230.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based Adaptive Structured Pruning for Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10865–10873, 2024. doi: 10.1609/aaai.v38i10.28960. URL https://ojs.aaai.org/index.php/AAAI/article/view/28960.

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1. Official model release, 2026. URL https://www.anthropic.com/claude-fable-and-mythos-5-1. Accessed September 23, 2026.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, second edition, 2006. doi: 10.1002/047174882X. URL https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Aritra Dutta and Somak Aditya. MuCRASP: Multimodal Chain-of-thought Reasoning aware Structured Pruning. *arXiv preprint arXiv:2605.25842*, 2026. URL https://arxiv.org/abs/2605.25842.

Elias Frantar and Dan Alistarh. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. SlimLLM: Accurate Structured Pruning for Large Language Models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2505.22689.

Junlin He, Yihong Tang, Tong Nie, Guilong Li, Binyu Yang, Jinxiao Du, Lijun Sun, and Wei Ma. Reasoning-preserved Efficient Distillation of Large Language Models via Activation-aware Initialization. *arXiv preprint arXiv:2605.29327*, 2026. URL https://arxiv.org/abs/2605.29327.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-Aware On-Policy Distillation of Language Models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.07079.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards Streamlined Distillation for Large Language Models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.03898.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.07067.

Steven Kolawole, Lucio Dery, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. Everybody Prune Now: Structured Pruning of LLMs with only Forward Passes. *arXiv preprint arXiv:2402.05406*, 2024. URL https://arxiv.org/abs/2402.05406.

Eldar Kurtic, Elias Frantar, and Dan Alistarh. ZipLM: Inference-Aware Structured Pruning of Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2302.04089.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016.

Gui Ling, Ziyang Wang, Yuliang Yan, and Qingwen Liu. SlimGPT: Layer-wise Structured Pruning for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/c1c44e46358e0fb94dc94ec495a7fb1a-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the Structural Pruning of Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/44956951349095f74492a5471128a7e0-Abstract-Conference.html.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. *arXiv preprint arXiv:2403.03853*, 2024. URL https://arxiv.org/abs/2403.03853.

Xiang Meng, Shibal Ibrahim, Kayhan Behdin, Hussein Hazimeh, Natalia Ponomareva, and Rahul Mazumder. OSSCAR: One-Shot Structured Pruning in Vision and Language Models with Combinatorial Optimization. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35354–35377, 2024. URL https://proceedings.mlr.press/v235/meng24a.html.

Ocean Monjur, Shahriar Kabir Nahin, and Anshuman Chhabra. Revisiting the Effectiveness of LLM Pruning for Test-Time Scaling. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. URL https://arxiv.org/abs/2604.25098.

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*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/4822991365c962105b1b95b1107d30e5-Abstract-Conference.html.

Hieu Trung Nguyen, Bao Nguyen, and Viet Anh Nguyen. Structured Pruning for Diverse Best-of-N Reasoning Optimization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 23911–23922, 2025. doi: 10.18653/v1/2025.findings-acl.1225. URL https://aclanthology.org/2025.findings-acl.1225/.

OpenAI. GPT-6 Astra. Official model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-6-astra. Accessed September 23, 2026.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing Verifiable Instruction Following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2507.02833. Datasets and Benchmarks Track.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

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 *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46136–46155, 2024. URL https://proceedings.mlr.press/v235/song24f.html.

Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, et al. LLM Pruning and Distillation in Practice: The Minitron Approach. *arXiv preprint arXiv:2408.11796*, 2024. URL https://arxiv.org/abs/2408.11796.

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.

Shengkun Tang, Oliver Sieberling, Eldar Kurtic, Zhiqiang Shen, and Dan Alistarh. DarwinLM: Evolutionary Structured Pruning of Large Language Models. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2502.07780.

Tycho F. A. van der Ouderaa, Markus Nagel, Mart van Baalen, Yuki M. Asano, and Tijmen Blankevoort. The LLM Surgeon. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.17244.

Ziyan Wang, Enmao Diao, Qi Le, Pu Wang, Guanchu Wang, Minwoo Lee, Shu-ping Yeh, and Li Yang. Think Before You Prune: Self-Reflective Structured Pruning for Reasoning Language Models. *arXiv preprint arXiv:2512.02185*, 2025. URL https://arxiv.org/abs/2512.02185.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/160adf2dc118a920e7858484b92a37d8-Abstract-Conference.html.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative Knowledge Distillation: Bridging the Teacher-Student Gap Through Interleaved Sampling. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.11325.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, 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*, 2024. URL https://arxiv.org/abs/2402.11187.

Qingyu Zhang, Qianhao Yuan, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, Ming Xu, and Jiarui Li. ShortOPD: Recovering Pruned LLMs with Short-to-Long On-Policy Distillation. *arXiv preprint arXiv:2607.13124*, 2026. URL https://arxiv.org/abs/2607.13124.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *arXiv preprint arXiv:2304.11277*, 2023. URL https://arxiv.org/abs/2304.11277.

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*, 2025. URL https://arxiv.org/abs/2406.10594.

A Experimental Details

We describe the models and channel-scoring adapters in Sections A.1–A.2, followed by the training data (Section A.3) and recovery settings (Section A.4). Sections A.5 and A.6 specify benchmark evaluation and fixed-prefix analysis.

A.1 Models and Pruned Architecture

We use Qwen3-8B with thinking enabled and Qwen3-4B-Instruct-2507 in non-thinking mode (Yang et al., 2025). Each student's corresponding unpruned checkpoint serves as its frozen teacher. Structured pruning removes channels from the gated FFN. For hidden input z and intermediate activation

$$a = \mathrm{SiLU}(W^{\mathrm{gate}} z) \odot W^{\mathrm{up}} z, \qquad o = W^{\mathrm{down}} a, \tag{6}$$

removing channel j removes row j of W^{gate} and W^{up} and column j of W^{down}. Attention, embeddings, normalization layers, and the language-model head are retained. Every layer uses the same retained width, aligned to a multiple of 128. Table 2 gives the exact widths; whole-model parameter reduction is smaller than FFN-width reduction because only FFNs are pruned.

A.2 Pruning Criteria and Calibration

The compared methods share the architecture in Section A.1. They differ in channel scores, rather than layerwise width allocation. For response r, let $a_{r,t,j}$ be channel j's activation at token t and $W^{\mathrm{down}}_{:,j}$ its output-weight column. We suppress the layer index below. For ReCal, define the weighted activation expectation

$$\mathbb{E}_w[f(a_j)] = \frac{1}{R} \sum_{r=1}^{R} \sum_{t \in \mathcal{R}_r} \widetilde{w}_{r,t} f(a_{r,t,j}), \tag{7}$$

where $\mathcal{R}_r$ contains valid response tokens and each response has total weight one. The ordinary adapters aggregate uniformly over their prompt calibration inputs. ReCal instead uses teacher responses and the damage weights in Equation 4.

Minitron. Following activation-based width pruning (Muralidharan et al., 2024; Sreenivas et al., 2024), the implemented adapter first averages absolute activations within each calibration sequence, then aggregates their squares across sequences:

$$s_j = \sqrt{\sum_r \left(\frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} |a_{r,t,j}| \right)^2}. \tag{8}$$

Here $\mathcal{T}_r$ is the baseline sequence's token set. ReCal replaces the inner mean with $\sum_{t \in \mathcal{R}_r} \widetilde{w}_{r,t} |a_{r,t,j}|$. Squaring remains outside the within-sequence aggregation; this is not equivalent to a global mean-absolute-activation score.

Wanda. Wanda combines weight magnitude and activation statistics (Sun et al., 2024). Our structured channel adapter, Wanda-SP, scores

$$s_j = \|W^{\mathrm{down}}_{:,j}\|_2 \sqrt{\mathbb{E}[a_j^2]}. \tag{9}$$

ReCal substitutes $\mathbb{E}_w[a_j^2]$ for the uniform second moment. This preserves the weight factor and channel grouping. It is a structured adaptation of Wanda, whose original formulation prunes individual weights.

FLAP. The reported FLAP adapter uses the WIFN score (An et al., 2024):

$$s_j = \|W^{\mathrm{down}}_{:,j}\|_2 \mathbb{E}[|a_j|]. \tag{10}$$

ReCal replaces the activation expectation with $\mathbb{E}_w[|a_j|]$. Although the implementation also records the WIFV statistic $\|W^{\mathrm{down}}_{:,j}\|_2^2 \,\mathrm{Var}(a_j)$, WIFV does not select the reported masks. We do not use FLAP's adaptive layerwise sparsity or bias compensation; fixed per-layer widths and the same recovery pipeline allow comparison of channel selection across all adapters.

Table 2: Retained FFN widths and whole-model parameter reduction. The original widths are 12,288 for 8B and 9,728 for 4B.

Model	Target reduction	Retained width	Actual reduction	Parameters
8B	15%	9,472	15.21%	6.945B
8B	25%	7,680	24.89%	6.152B
8B	35%	5,760	35.26%	5.303B
4B	15%	7,552	14.96%	3.421B
4B	25%	6,144	24.63%	3.032B
4B	35%	4,608	35.19%	2.607B

LLM-Pruner. The first-order LLM-Pruner adapter groups the gate, up, and down weights associated with each channel, denoted by G_j, and uses the first-order score (Ma et al., 2023)

$$s_j = \sum_{u \in G_j} \left| u \frac{\partial \mathcal{L}}{\partial u} \right|. \tag{11}$$

The baseline uses next-token cross entropy on calibration prompts. RECAL differentiates a weighted response-token loss, $\mathcal{L} = \sum_t w_t \, \mathrm{CE}_t \, / \sum_t w_t$, and applies the same group score. Only FFN projection parameters receive gradients during scoring. The maximum scoring sequence length is 2,048 for both variants. Weights enter the loss before differentiation and before the absolute-value aggregation.

Calibration data and divergence. Both calibration pools are balanced equally across mathematics, code, science, and instruction following. The baseline pool contains prompts, whereas RECAL uses responses from the unpruned teacher, with prompt and padding tokens excluded from weighted statistics. The probe is the corresponding baseline pruned to the same target widths, without recovery training. Final scores are recomputed on the original unpruned model; probe weights and masks are not inherited by the final student.

For efficient damage estimation, let $S_{r,t}$ be the teacher's Top-k token set at a shared prefix. We retain the original probabilities on these tokens and collect the remaining mass in a single bucket:

$$\widehat{d}_{r,t} = \sum_{v \in S_{r,t}} p_T(v) \log \frac{p_T(v)}{p_P(v)} + p_T(\mathrm{other}) \log \frac{p_T(\mathrm{other})}{p_P(\mathrm{other})}, \tag{12}$$

where $p_M(\mathrm{other}) = 1 - \sum_{v \in S_{r,t}} p_M(v)$ for $M \in \{T, P\}$. We use $k = 16$ and a maximum calibration length of 8,192 tokens for the activation-based adapters. This is KL on a shared coarsened vocabulary, not KL after renormalizing only the Top-k tokens. The per-response weights are $\widetilde{w}_{r,t} = \widehat{d}_{r,t} / \sum_u \widehat{d}_{r,u}$. If the denominator is at most 10^{-9}, we use uniform weights over valid response tokens. Each response therefore has equal total weight regardless of its length. Damage scores are aligned with the response-token activations that predict the affected next token before statistics are accumulated. These comparisons evaluate the combined use of teacher trajectories and damage weighting.

A.3 TRAINING DATA

SFT and OPD use separate prompt pools drawn from mathematics, code, science, and instruction following, initially mixed in equal proportions (1:1:1:1). Table 3 lists the sources. Science prompts balance physics, chemistry, and biology in a 1:1:1 ratio; OPD instruction prompts balance one through five constraints. Mathematics prompts added from the SFT source use a separate source partition, and the final SFT and OPD pools have no normalized-prompt overlap. Prompts undergo deduplication and benchmark-overlap screening.

Only prompts are retained from these sources. For SFT, the corresponding unpruned teacher generates new targets with temperature 0.6 and top-p 0.95. Truncated or malformed responses are excluded, so the final SFT response mixture is not exactly domain-balanced. All compared methods for a given model share this same target set. Mathematics and science use a common final-answer format, code follows the LiveCodeBench Python template, and instruction-following prompts retain their original constraints. OPD uses prompts only; it does not consume source answers or external task rewards.

Table 3: Public prompt sources. Dataset identifiers are linked to their source pages. NVIDIA datasets carry the `nvidia/` namespace; IF-multi is from `allenai/`. The initial domain mixture is 1:1:1:1 in both stages.

Domain	SFT prompt source	OPD prompt source
Math	Nemotron-SFT-Math-v4	Nemotron-RL-Math-v2; disjoint supplement from SFT-Math-v4
Code	Nemotron-SFT-Competitive-Programming-v2	Nemotron-RL-coding-competitive_coding
Science	Nemotron-SFT-Science-v2	Nemotron-RL-Science-v1
Instruction	Nemotron-SFT-Instruction-Following-Chat-v3	IF_multi_constraints_upto5

A.4 Recovery Hyperparameters

Table 4 summarizes the settings that determine training and sampling. Both stages use AdamW (Loshchilov & Hutter, 2019). SFT minimizes next-token cross entropy on assistant responses, including reasoning tokens when thinking is enabled. User and system tokens do not contribute to the loss. OPD then samples fresh responses from the current student and evaluates the teacher at those student-generated prefixes. All completed main runs use 200 OPD updates.

Restricted-support OPD signal. Equation 2 describes the full-vocabulary reference objective. The implemented signal instead uses the student's Top-16 token IDs K_t at each prefix h_t. Student probabilities are normalized on this support:

$$\widetilde{p}_S(v \mid h_t) = \frac{p_S(v \mid h_t)}{\sum_{u \in K_t} p_S(u \mid h_t)}, \qquad v \in K_t. \tag{13}$$

The token-level teacher signal is

$$r_{t,v} = -\widetilde{p}_S(v \mid h_t)\big[\log p_S(v \mid h_t) - \log p_T(v \mid h_t)\big]. \tag{14}$$

The implementation uses this signal directly as an advantage in its clipped policy update, without GAE or an external task reward. The log probabilities retain their original vocabulary normalization. Thus, this is a restricted-support reverse-KL-style signal, not the full KL in Equation 2, and it differs from the teacher-support-plus-residual calculation used for calibration.

Shared optimization conditions. Paired runs start from the same unpruned checkpoint, retain identical per-layer widths, and share the same SFT targets, OPD prompts, data ordering, and optimization settings. Training uses FSDP (Zhao et al., 2023), BF16 forward computation, and FP32 master weights. The recorded environment uses eight NVIDIA H20 GPUs. Matching update counts does not imply identical sampled token counts or GPU time, since students can generate different response lengths.

A.5 Benchmarks and Evaluation

We evaluate every available checkpoint before recovery, after SFT, and after SFT plus OPD, using the same response budget of 20,480 tokens. AIME is sampled at temperature 0.6 and top-p 0.95; the other benchmarks use greedy decoding. Metrics are defined below.

AIME 2024–2026. Each annual set contains competition mathematics problems with exact numerical answers. We average correctness over eight sampled responses per question and then average the three yearly scores. This is average sampled accuracy, not pass@8. The final boxed answer is extracted and normalized before comparison.

GPQA-Diamond. GPQA evaluates graduate-level science questions in biology, chemistry, and physics (Rein et al., 2023). We use the Diamond test split and report multiple-choice accuracy, extracting the final A/B/C/D answer from the common boxed-answer format.

IFBench. IFBench evaluates compliance with verifiable instructions (Pyatkin et al., 2025). We use the official test set and verifier and report instruction-level strict accuracy (IFB-S). A response can satisfy some constraints while violating others, so this metric is distinct from prompt-level strict accuracy.

Table 4: Recovery settings shared by each Base/ReCal pair. SFT is offline distillation on teacher responses. OPD uses student rollouts.

Setting	SFT	OPD
Optimizer	AdamW	AdamW
Learning rate	10^{-5}	10^{-6}
Schedule	Cosine, 5% warmup	Constant
Batch size	32 sequences	64 prompts
Training budget	2 epochs	200 updates
Responses per prompt	1 teacher target	4 student rollouts
Sampling temperature	0.6 (teacher)	1.0 (student)
Sampling top-p	0.95	1.0
Response limit	20,480	20,480
Sequence / prompt limit	21,504 sequence	2,048 prompt
Distillation support	Sampled teacher tokens	Student Top-16
Teacher temperature	—	1.0
Repetition penalty	—	1.05
Optimization passes per update	—	1
Policy clip ratio	—	0.2
Gradient clipping	1.0	1.0
Weight decay	0	0

LiveCodeBench. We use the code-generation task in LiveCodeBench-v5, covering problems released from October 2024 through February 2025 (Jain et al., 2025). We report pass@1 under the official input/output or function-call templates, with a 10-second execution timeout per problem.

Reporting conventions. Scores are percentages. Within each main-table model block, boldface marks the highest displayed score in each column and underlining the second-highest distinct score, across all pruned stages, excluding the unpruned reference. Ties receive the same formatting. The final Avg. column is the unweighted arithmetic mean of AIME24, AIME25, AIME26, GPQA, IFBench, and LiveCodeBench-v5. AIME averages elsewhere use only the three AIME benchmarks. The original seven paired gains were reported before rounding and can differ by 0.1 percentage point from displayed-score differences. The additional 8B LLM-Pruner gain is computed from the reported three-decimal scores. Recovery-stage averages use full-precision repository records for all eight pairs; final-performance comparisons also use all eight. The unpruned 8B AIME/GPQA reference was regraded at the common response limit; IFBench and LiveCodeBench use fresh generations. Individual-year AIME scores are available for every main-table checkpoint and both LLM-Pruner OPD pairs. The 8B LLM-Pruner + ReCal Pruned/SFT scores are included in Table 5.

A.6 Fixed-Prefix Diagnostic Protocol

We use held-out teacher responses balanced across the four domains to measure distributional damage. Both teacher and student evaluate the same prefix at every response token. Within each response, tokens are ranked by raw-pruned baseline KL and divided into five equal-count groups, Q1–Q5, from lowest to highest damage. The groups are fixed thereafter; no checkpoint selects its own favorable token subset.

Concentration curves average each response's fraction of damage carried by its highest-damage tokens. The fixed-group plots first average KL within each response/group, then across responses. For the Q5 damage share, each response's group-average KL is multiplied by its token count, divided by total response damage, and then averaged across responses. This is an average of per-response ratios, not a ratio of group means.

The diagnostic pool contains 128 responses per model, equally split across domains, with no filtering by answer correctness or damage. Recovery tracking uses 8B Wanda at the main compression ratio. Section 3 uses only baseline checkpoints; Section 5.5 adds the calibrated counterparts on the same responses and token groups. Intervals use 2,000 trajectory-bootstrap resamples; differences use paired resampling. Tokens within a response are not treated as independent samples.

B SUPPLEMENTARY EXPERIMENTS

We supplement the main results with the LLM-Pruner integration in Section B.1, additional damage analysis in Section B.2, and generation diagnostics in Section B.3. Section B.4 documents the introduction diagnostics, and Section B.5 reports benchmark results not displayed in the main paper.

B.1 LLM-PRUNER INTEGRATION

Figure 8 extends the comparison to both LLM-Pruner models. In 4B, post-OPD AIME, GPQA, and code improve, while IFBench declines slightly. In 8B, AIME falls from 12.4% to 9.2% and IFBench from 22.4% to 19.5%, whereas GPQA rises from 34.8% to 44.9% and code from 17.5% to 28.3%. The 8B AIME24/25/26 scores change from 18.8/7.9/10.4% to 13.8/5.0/8.7%, so the mathematics decline occurs in every year. Thus, the positive 4B result does not generalize uniformly to 8B. The 8B mathematics deficit is already present after SFT: 3.2% for the calibrated student versus 7.9% for Base. OPD narrows this deficit from 4.7 to 3.2 points but does not reverse it. As with the other adapters, the comparison changes both calibration context and weighting.

B.2 ADDITIONAL DAMAGE ANALYSIS

Figure 9 adds the 4B concentration curves and paired KL changes before and after SFT. The concentration pattern extends to 4B. In 8B Wanda, the paired changes reveal increased discrepancy in some lower-damage groups after pruning and SFT, despite the reduction in Q5. These earlier stages complement the post-OPD comparison in Figure 7(b).

B.3 GENERATION AND ALIGNMENT DIAGNOSTICS

Figure 10(a) reports net completion rescue on paired AIME generations: the number where Base truncates but RECAL completes, minus the number with the reverse outcome. Pairing uses the same question and sample index, not an identical sampled trajectory. All seven post-OPD pairs with completion diagnostics are positive, ranging from 12 to 116 net completions per 720 generations. However, 4B Wanda and FLAP initially have fewer completed generations immediately after pruning. Completion after pruning is therefore an incomplete indicator of the eventual recovered model. Its aggregate post-OPD truncation rate nevertheless increases from 76.8% to 79.4% with RECAL, consistent with its mathematics exception.

The three 8B comparisons enter OPD with 1.9–3.6 percentage points higher teacher–student Top-16 overlap and finish 1.0–2.9 points higher. The 4B changes are smaller and not uniformly positive. Because these statistics use each student's own rollouts, they describe the trajectories encountered during recovery and do not isolate agreement at matched states.

Trajectory-level diagnostics further distinguish response length from token damage. Across probes, response length correlates strongly with summed KL (Spearman 0.783–0.886), motivating equal total weight per response. Cross-probe mean-damage rankings are highly correlated (0.963–0.975), so the necessity of a criterion-specific probe is not established. Instruction responses have high mean damage, yet IFBench does not consistently improve. A high divergence is therefore not interchangeable with task importance.

B.4 INTRODUCTION FIGURE MEASUREMENT

Figure 1(b,c) uses the measured Qwen3-8B Minitron, Wanda, and FLAP baselines at the main compression ratio. Panel (b) reports mean AIME accuracy across 2024–2026 for the unpruned teacher, raw-pruned student, and post-SFT student. The shared teacher score is 61.4%. Panel (c) reports the fraction of AIME generations reaching the common response-length limit after SFT and after OPD. Scores follow the sampling and decoding settings in Section A.5.

Truncation measures failure to finish within the evaluation budget; it does not distinguish repetition, an unfinished derivation, or a completed answer followed by excess text. It is therefore reported separately from answer accuracy. These aggregate measurements do not estimate reverse KL conditional on failure type. The introduction's discussion of teacher confirmation at flawed prefixes draws on

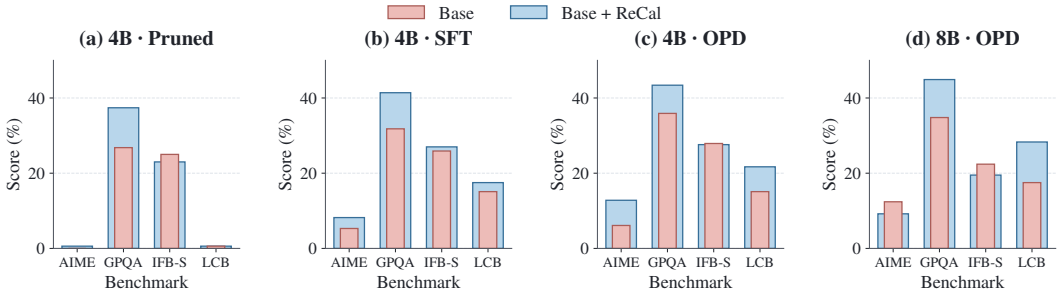


Figure 8: **LLM-Pruner with and without RECAL.** (a–c) The three 4B recovery stages. (d) The completed 8B OPD pair. AIME denotes the three-year average. Wide blue bars show Base + RECAL; narrow red bars show Base.

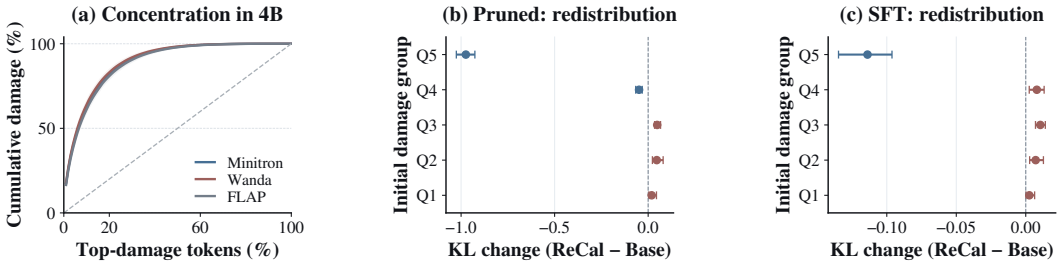


Figure 9: **Additional concentration and redistribution results.** (a) Within-response damage concentration in 4B. (b,c) Paired RECAL minus Base KL after pruning and SFT. Groups remain fixed by the raw-pruned baseline. Intervals are 95% paired trajectory-bootstrap intervals.

prior work (Zhang et al., 2026); our fixed-prefix experiments in Section 3 examine pruning damage directly.

B.5 ADDITIONAL BENCHMARK RESULTS

Tables 5 and 6 report the calibrated Minitron, FLAP, and additional LLM-Pruner scores not displayed in Table 1 or Figure 4. The first table adds Pruned/SFT checkpoints; the second adds post-OPD science and instruction scores. Matching baseline scores are in Table 1. The 4B/8B LLM-Pruner comparisons are shown in Figure 8. Complete numerical records, including the ratio sweep, are retained in the accompanying data files.

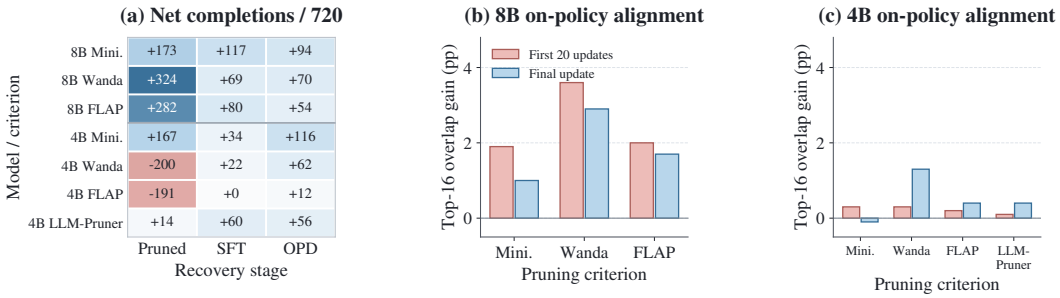


Figure 10: **Supplementary recovery diagnostics.** (a) Net completion rescue across the three stages. (b,c) Top-16 teacher–student overlap gains during the first 20 OPD updates and at the final update. Each model generates its own trajectories.

Table 5: Additional Pruned and SFT scores at 25% parameter reduction. These calibrated variants are absent from the main table; their post-OPD scores appear in Figures 4 and 8. Scores are percentages.

Method	Stage	AIME24	AIME25	AIME26	GPQA	IFB-S	LCB-v5
Qwen3-8B							
Minitron + ReCal	Pruned	4.6	0.8	1.7	30.8	18.6	5.4
	SFT	25.8	16.7	18.3	35.4	18.6	17.5
FLAP + ReCal	Pruned	15.0	9.6	7.5	28.8	18.6	8.4
	SFT	41.2	30.8	32.1	38.4	20.1	21.1
LLM-Pruner + ReCal	Pruned	0.0	0.0	0.0	27.3	18.0	0.0
	SFT	6.3	0.4	2.9	41.9	17.2	19.3
Qwen3-4B-Instruct							
Minitron + ReCal	Pruned	4.2	1.7	3.7	30.3	23.3	6.6
	SFT	22.9	15.0	15.8	37.9	25.6	16.3
FLAP + ReCal	Pruned	14.6	10.0	8.3	32.8	25.3	11.4
	SFT	27.9	23.7	18.3	34.8	24.4	18.7

Table 6: Additional post-OPD science and instruction results at 25% reduction, complementing the AIME and code results in Figure 4. Scores are percentages.

Model	Method	GPQA	IFB-S
8B	Minitron + ReCal	36.4	19.5
8B	FLAP + ReCal	37.4	19.8
4B	Minitron + ReCal	36.4	27.0
4B	FLAP + ReCal	35.4	27.0

C Forward KL and Teacher-Trajectory Preservation

We first relate full-vocabulary forward KL to preservation of teacher sequence probabilities in Section C.1. Section C.2 explains why this connection motivates the method without guaranteeing recovery gains.

C.1 Sequence-Distribution Connection

For a fixed prompt x, write P_T and P_m for teacher and pruned-model sequence distributions up to a shared horizon H, with an absorbing EOS state. Define

$$D(m) = \mathbb{E}_{y\sim P_T} \sum_{t=1}^{H} \mathrm{KL}\big(p_T(\cdot \mid x, y_{<t}) \| p_m(\cdot \mid x, y_{<t})\big). \tag{15}$$

Proposition 1 (Teacher-trajectory preservation)**.** *Assume P_m is positive wherever P_T is positive. For any set A of completed responses,*

$$\mathrm{KL}(P_T \| P_m) = D(m), \qquad P_m(A) \geq P_T(A) - \sqrt{D(m)/2}. \tag{16}$$

Proof. Both distributions use the same vocabulary and stopping convention. The autoregressive factorization and the KL chain rule (Cover & Thomas, 2006) give

$$\mathrm{KL}(P_T \| P_m) = \mathbb{E}_{y\sim P_T} \log \frac{\prod_{t=1}^{H} p_T(y_t \mid x, y_{<t})}{\prod_{t=1}^{H} p_m(y_t \mid x, y_{<t})} \tag{17}$$

$$= \sum_{t=1}^{H} \mathbb{E}_{y_{<t}\sim P_T} \mathrm{KL}\big(p_T(\cdot \mid x, y_{<t}) \| p_m(\cdot \mid x, y_{<t})\big) = D(m). \tag{18}$$

Terms after EOS are zero. With natural logarithms, Pinsker's inequality yields

$$|P_T(A) - P_m(A)| \leq \mathrm{TV}(P_T, P_m) \leq \sqrt{\mathrm{KL}(P_T \| P_m)/2}, \tag{19}$$

which proves the claim. The same argument applies to a shared distribution over prompts by considering the joint prompt–response distribution. This proposition applies standard information-theoretic identities to the pruning setting; it is not a new KL inequality. □

C.2 Interpretation and Scope

The connection explains why teacher-generated prefixes and teacher-weighted disagreement are relevant to retaining teacher-supported continuations. It does not show that KL weighting directly minimizes $D(m)$, that smaller bounds order actual task accuracies, or that SFT and OPD preserve any ordering of masks. The teacher may also generate incorrect answers.

The population quantity $D(m)$ sums token divergences before averaging over responses. The fixed-group means and length-normalized response averages in our diagnostics are different statistics. Moreover, merging tail tokens into the residual bucket in Equation 12 can only decrease KL by the data-processing inequality. The implemented coarsened KL is thus not an upper bound on full-vocabulary damage and cannot be substituted into Proposition 1 to certify sequence preservation. Recovery benefits are established empirically through the paired experiments.

D Additional Related-Work Comparisons

We extend Section 6 by distinguishing the structural decisions, calibration signals, and recovery interventions in prior work. These distinctions determine which methods are direct comparisons and which address complementary problems.

Pruning granularity and selection criteria. SparseGPT removes individual weights using an approximate second-order reconstruction procedure, while Wanda combines weight magnitude with input activations (Frantar & Alistarh, 2023; Sun et al., 2024). Their original weight-level

formulations differ from removing complete FFN channels; our Wanda-SP adaptation is specified in Appendix A.2. Among structured methods, the LLM Surgeon uses curvature information and weight updates, OSSCAR formulates layer reconstruction as combinatorial optimization, and SlimGPT applies batched greedy pruning within an Optimal Brain Surgeon framework (van der Ouderaa et al., 2024; Meng et al., 2024; Ling et al., 2024). Bonsai estimates importance through forward-pass perturbations, avoiding backward passes (Kolawole et al., 2024). These methods primarily differ in how they estimate or optimize the cost of deleting a structure. RECAL changes the token evidence supplied to a criterion. Compatibility with the four evaluated adapters does not establish compatibility with every reconstruction or curvature-based method.

Layer and block methods offer a different compression axis. ShortGPT ranks layer redundancy, LaCo collapses layers, BlockPruner distinguishes attention and feed-forward blocks, and SLEB eliminates redundant transformer blocks (Men et al., 2024; Yang et al., 2024; Zhong et al., 2025; Song et al., 2024). SliceGPT rotates representations before removing rows and columns (Ashkboos et al., 2024). ZipLM incorporates inference cost into structural decisions (Kurtic et al., 2023). Our experiments retain all layers and attention modules and compare FFN channels at identical widths. Consequently, they do not establish the best compression axis or measured deployment speedups.

Accounting for recovery during compression. Sheared LLaMA combines targeted structured pruning with continued pretraining; Minitron studies pruning and distillation across multiple architectural dimensions (Xia et al., 2024; Muralidharan et al., 2024; Sreenivas et al., 2024). More directly, DarwinLM incorporates brief candidate training into evolutionary architecture search (Tang et al., 2026). NIRVANA uses function-space saliency to account for fine-tuning dynamics and KL-based calibration-data selection (Ai et al., 2026). These works already establish that compression should consider subsequent training. Our contribution is the specific teacher–probe token signal and its integration into calibration for an unchanged SFT–OPD pipeline, rather than the general idea of recovery-aware pruning. Unlike candidate training or layer-budget search, RECAL fixes the target widths and uses an unrecovered probe to redistribute calibration weight within responses.

Preserving reasoning. Reasoning models can exploit long generated traces and test-time computation (DeepSeek-AI et al., 2025; Yang et al., 2025), making next-token fit alone an incomplete evaluation of compression. RESP uses self-generated reasoning traces and decode-only gradient scoring, SPRINT considers diverse best-of-N reasoning, and recent evaluations examine pruning under test-time scaling (Wang et al., 2025; Nguyen et al., 2025; Monjur et al., 2026). RED uses activation-aware initialization for reasoning-model distillation (He et al., 2026). MuCRASP extends reasoning-aware pruning to multimodal models using reasoning and modality signals (Dutta & Aditya, 2026). Our fixed-prefix analysis instead tracks baseline-defined damaged tokens through recovery. It supports an empirical calibration strategy, without showing that large token divergence always identifies task-critical reasoning steps. Our mathematics and code gains, instruction-following losses, and the 8B LLM-Pruner exception bound that interpretation.

Distillation objectives and rollout distributions. Classical distillation transfers teacher predictions, while sequence-level distillation fits generated teacher responses (Hinton et al., 2015; Kim & Rush, 2016). Learning on learner-induced states addresses a general distribution-shift problem in sequential prediction (Ross et al., 2011); language-model OPD obtains teacher supervision on these states (Agarwal et al., 2024; Gu et al., 2024). Speculative distillation interleaves student proposals and teacher replacements to balance on-policy coverage with response quality (Xu et al., 2025). Self-Distilled Reasoner obtains supervision from a solution-conditioned version of the same model (Zhao et al., 2026). These supervision designs differ from our frozen unpruned teacher. Entropy-aware OPD also employs forward KL, but uses teacher uncertainty to adapt the recovery objective (Jin et al., 2026). RECAL uses teacher–probe disagreement before recovery to weight structural calibration. Thus, the divergence direction alone is not the novelty; its role in pre-recovery structure selection is the relevant distinction.