PRUNE WHAT CAN BE RECONSTRUCTED: CARVE FOR COMPENSATION-AWARE NEURON SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper focuses on post-training structured pruning of MLP hidden neurons in pretrained vision backbones, without fine-tuning. Most modern approaches follow the same paradigm: they first rank neurons using a fixed importance score and then compensate for the neurons that are removed. We propose a different paradigm: neurons should be selected according to *what cannot be recovered after compensation*, rather than according to their importance before compensation. Following this principle, we introduce Covariance-Aware Reduction Via Elimination (CARVE), which greedily removes the neuron that causes the smallest increase in reconstruction error after optimal affine compensation through the surviving neurons. To the best of our knowledge, CARVE is the first post-training structured pruning method whose selection criterion is directly defined by the exact error remaining after optimal compensation and is updated as the set of surviving neurons changes. We further introduce a global allocation strategy that rescales local errors by the residual-stream signal for cross-block comparison. We evaluate CARVE on ImageNet-1k across nine vision backbones, from 5.7M to 632M parameters, and nine MLP sparsities from 10% to 90%. Without fine-tuning, CARVE outperforms the strongest competing baseline by **5.6 Top-1 accuracy points on average over the nine vision backbones** at sparsities of 50% and above, with gains reaching **20.7 points** at aggressive compression.

1 INTRODUCTION

Vision Transformers are powerful but expensive to deploy, as their memory and compute grow rapidly with width and depth (Dosovitskiy et al., 2021; Touvron et al., 2021; 2022). Compression methods include quantisation (Jacob et al., 2018), distillation (Hinton et al., 2015), token reduction (Bolya et al., 2023), and pruning. We focus on *structured pruning*, which removes complete units, resulting in smaller dense models that require no specialised sparse kernels.

A Transformer is a sequence of Transformer blocks, each containing a self-attention module and an MLP. The MLPs hold roughly two thirds of a Transformer's parameters, so their hidden neurons are natural targets for structured pruning (Zhang & Yang, 2026; Berisha et al., 2025). Pruning them raises two questions: *which neurons should be removed within each block?* and *how should a global pruning budget be distributed across blocks?* We study these questions in a *post-training* setting, using only a small calibration set and no fine-tuning or post-pruning recovery (Frantar & Alistarh, 2022; Zhang & Yang, 2026).

Most pruning methods follow an *importance-first* paradigm, ranking neurons using magnitude (Han et al., 2015), loss sensitivity (Lee et al., 2019), or activation and weight statistics (Sun et al., 2024; Shen et al., 2025), and simply removing those deemed least important.

More recent approaches additionally compensate for what is removed, an idea rooted in Optimal Brain Surgeon (Hassibi & Stork, 1992) and later adapted to post-training compression (Frantar & Alistarh, 2022; 2023). For structured pruning, FLAP pairs a fluctuation-based criterion with an additive bias compensation (An et al., 2024), and VBP applies the same pairing to MLP neurons, folding the removed post-activation mean into the second layer's bias (Berisha et al., 2025). This introduced an important new paradigm: pruning should not only decide *what to remove*, but also *how to recover what was removed.*

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.

Recent methods push this compensation paradigm further. GRAIL (Tang et al., 2026) and CORP (Zhang & Yang, 2026) reconstruct removed representations from the surviving neurons, while DVBP (Lee et al., 2026) combines reconstruction with a denoised variance criterion. However, these methods still largely follow the same strategy: *selection is performed before compensation*. The score decides which neurons to remove, while reconstruction is applied only afterwards. Thus, the selection criterion does not directly measure the error that will actually remain after compensation.

This limitation appears at two levels. **Locally**, importance does not measure recoverability: a neuron may appear important in isolation yet be accurately reconstructed from the remaining neurons, whereas a less prominent neuron may carry information that cannot be recovered after pruning. **Globally**, reconstruction errors cannot be directly compared across Transformer blocks: blocks operate at different scales, and an error introduced early is carried, and can be amplified, by every block that follows. Structured pruning must therefore answer both *which neuron causes the least irrecoverable error?* and *in which block is such an error least harmful?*

Research question. We ask a simple question: *which neurons should be removed when accounting for both optimal local compensation and global allocation across blocks?* This leads to a new pruning paradigm, in which selection is defined by what survives compensation. Following this principle, we introduce Covariance-Aware Reduction Via Elimination (CARVE). Within each MLP block, CARVE removes the neuron whose deletion causes the smallest increase in block output error, once the surviving neurons have absorbed its contribution through the optimal affine compensation. This cost is recomputed as the survivor set evolves, rather than fixed before pruning. Closed-form rank-one updates make the resulting greedy elimination efficient and practical. CARVE is the first post-training structured pruning method whose selection criterion is defined by the exact marginal error remaining after its own optimal compensation. CARVE also addresses *where* to prune. Since local reconstruction errors are not directly comparable across blocks, we calibrate them relative to the residual signal. This yields a common criterion for allocating a single pruning budget across the network.

Our contributions are threefold. **(i) A compensation-aware pruning principle.** To the best of our knowledge, we are the first to formally ground structured neuron pruning in the error that remains *after optimal compensation*. This cost adapts after every removal as the surviving neurons change; we derive it in closed form together with efficient rank-one updates. **(ii) A residual-aware global allocation.** We show that local post-compensation errors are not directly comparable across blocks, especially at high sparsity. We account for the residual-stream scale of each error to derive a common criterion for global pruning allocation. **(iii) A controlled and broad empirical validation.** Through controlled ablations, we disentangle the effects of neuron selection and cross-block allocation. We show that compensation-aware selection consistently improves pruning, while allocation becomes increasingly important as sparsity grows. We evaluate CARVE on nine vision backbones from 5.7M to 632M parameters on ImageNet-1k and CIFAR-100 across sparsities from 10% to 90%. Without fine-tuning, distillation, or post-pruning recovery, CARVE improves over the strongest competing baseline by **5.6 Top-1 points on average** on the nine vision backbones at sparsities of 50% and above, with gains up to **20.7 points**.

2 Related work

Neuron selection. Structured pruning has used weight magnitude (Han et al., 2015; Li et al., 2017), linear independence (Shen et al., 2025), first- and second-order loss sensitivity (Lee et al., 2019; LeCun et al., 1989), and activation-weight statistics (Sun et al., 2024); several of these were proposed for unstructured pruning and transfer to neurons by summing the per-weight score over the group. More recent methods use activation variance (An et al., 2024; Berisha et al., 2025; Lee et al., 2026) or activation energy combined with outgoing weights (Zhang & Yang, 2026). These criteria are computed before pruning and do not directly measure the error that remains after reconstructing a removed neuron.

Neuron compensation. Compensating removed parameters goes back to Optimal Brain Surgeon (Hassibi & Stork, 1992), and was later developed for post-training compression by OBC (Frantar & Alistarh, 2022) and SparseGPT (Frantar & Alistarh, 2023). For structured pruning, FLAP and VBP use bias or mean compensation (An et al., 2024; Berisha et al., 2025), while GRAIL, CORP,

and DVBP reconstruct removed activations from the surviving ones (Tang et al., 2026; Zhang & Yang, 2026; Lee et al., 2026). For a fixed kept set, CARVE uses the same centred affine reconstruction as CORP; its novelty is to use the resulting reconstruction error itself to decide *which neuron to remove next*.

Cross-block allocation. VBP and DVBP globally pool neuron scores (Berisha et al., 2025; Lee et al., 2026), FLAP standardises scores before allocation (An et al., 2024), while GRAIL uses uniform layer-wise pruning (Tang et al., 2026) and CORP does the same (Zhang & Yang, 2026). CARVE instead compares the post-compensation errors across blocks after normalising them by the residual-stream scale, making the pruning budget consistent with the error introduced at each block.

Greedy elimination. Greedy backward elimination is generally not globally optimal, though optimality and approximation guarantees exist under additional assumptions (Couvreur & Bresler, 2000; Das & Kempe, 2011; Elenberg et al., 2018). The within-block problem CARVE solves is an output-weighted column subset selection problem (Shitov, 2021; Deshpande et al., 2006), combinatorial in general. Our contribution is not a new guarantee but an exact, cheap evaluation of each greedy step for this objective.

3 METHODOLOGY

In this section, we introduce CARVE, a post-training structured pruning method for MLP neurons. As motivated in Section 1, CARVE addresses two questions: *which neuron should be removed within each Transformer block?* and *how should the global pruning budget be distributed across blocks?* Locally, we select neurons according to the error that remains after optimal compensation. Globally, we make these local errors comparable across blocks by normalising them against the residual-stream signal they perturb. All notations are listed in Appendix A.

3.1 COMPENSATION-AWARE NEURON SELECTION

Local pruning problem. Consider Transformer block $b \in \{1, \dots, B\}$ with embedding dimension D^b and MLP hidden width H^b. For an MLP input $x^b \in \mathbb{R}^{D^b}$, we define[1]

$$h^b = \phi(W_1^b x^b + c_1^b) \in \mathbb{R}^{H^b}, \qquad y^b = W_2^b h^b + c_2^b \in \mathbb{R}^{D^b}, \tag{1}$$

where $W_1^b \in \mathbb{R}^{H^b \times D^b}$ and $c_1^b \in \mathbb{R}^{H^b}$ are the weight and bias of the first linear layer; ϕ is the elementwise activation, and $W_2^b \in \mathbb{R}^{D^b \times H^b}$, $c_2^b \in \mathbb{R}^{D^b}$ those of the second. We estimate from the calibration set

$$\mu_b = \mathbb{E}[h^b], \qquad \Sigma_b = \mathrm{Cov}(h^b). \tag{2}$$

We omit the block index in the remainder of this subsection.

Let $K \subseteq \{1, \dots, H\}$ be the k neurons that are kept and $P = K^c$ those that are removed. Structured pruning partitions both the hidden representation and the corresponding columns of the second linear layer:

$$h = \begin{bmatrix} h_K \\ h_P \end{bmatrix}, \qquad W_2 = [W_{2,K} \quad W_{2,P}], \tag{3}$$

so that

$$y = W_{2,K} h_K + W_{2,P} h_P + c_2. \tag{4}$$

After pruning, only h_K remains, so the block can produce only outputs of the form

$$\widetilde{y} = \widetilde{W}_2 h_K + \widetilde{c}_2, \tag{5}$$

with free parameters $\widetilde{W}_2 \in \mathbb{R}^{D \times k}$ and $\widetilde{c}_2 \in \mathbb{R}^D$: choosing them is the compensation. We seek the kept neurons and compensated parameters that best preserve the original MLP output:

$$\varepsilon_k^\star = \min_{\substack{|K|=k \\ \widetilde{W}_2, \widetilde{c}_2}} \mathbb{E}\left[\|y - \widetilde{y}\|_2^2\right]. \tag{6}$$

[1]Two architectural variants are treated in Appendix E: a learned gain applied after the MLP, such as LayerScale, which is absorbed into the effective output weights; and gated MLPs such as SwiGLU, where h is the gated product and one index set slices both input projections.

Optimal compensation for a fixed set. For a fixed K, the missing contribution $W_{2,P}h_P$ can be recovered whenever the removed activations can be predicted from the kept ones. Partitioning the covariance as

$$\Sigma = \begin{pmatrix} \Sigma_{KK} & \Sigma_{KP} \\ \Sigma_{PK} & \Sigma_{PP} \end{pmatrix}, \tag{7}$$

the optimal affine prediction of the removed activations from the survivors is obtained by regressing the removed activations h_P on the kept ones h_K. Hence we have

$$h_P \approx \mu_P + \beta^\top (h_K - \mu_K), \qquad \beta = \Sigma_{KK}^{-1}\Sigma_{KP}. \tag{8}$$

Substituting this prediction into equation 4 absorbs the missing contribution directly into the surviving weights.

Proposition 1 (Optimal affine compensation)**.** *For a fixed kept set K with $\Sigma_{KK} \succ 0$, the optimal compensated parameters are*

$$\widetilde{W}_2 = W_{2,K} + W_{2,P}\beta^\top, \qquad \widetilde{c}_2 = c_2 + W_{2,P}\mu_P - W_{2,P}\beta^\top \mu_K. \tag{9}$$

The proof is given in Appendix C. The compensation preserves the output mean exactly. The mean compensation of VBP (Berisha et al., 2025) corresponds to $\beta = 0$.

After optimal compensation, the remaining error is

$$\varepsilon(K) = \mathrm{Tr}\left(W_{2,P}\Sigma_{P|K}^{\mathrm{res}} W_{2,P}^\top\right), \tag{10}$$

where

$$\Sigma_{P|K}^{\mathrm{res}} = \Sigma_{PP} - \Sigma_{PK}\Sigma_{KK}^{-1}\Sigma_{KP} \tag{11}$$

is the part of the removed representation that cannot be recovered from h_K. Finding the best K is an output-weighted column subset selection problem and is NP-hard in general (Shitov, 2021); see Appendix F. We therefore use greedy backward elimination.

Selecting after compensation. Rather than ranking neurons first and compensating afterwards, CARVE uses the error that remains *after* optimal compensation to decide what to remove. Let K^t be the surviving neurons after t removals and

$$\Omega^t = \Sigma_{K^t K^t}^{-1}. \tag{12}$$

As demonstrated in Appendix C, we have, for each $i \in K^t$,

$$\frac{1}{\Omega_{ii}^t} = \min_{\alpha} \mathrm{Var}\left(h_i - \alpha^\top h_{K^t \setminus \{i\}}\right), \tag{13}$$

which measures the part of neuron i that cannot be reconstructed from the other surviving neurons.

Let $w_i^t \in \mathbb{R}^D$ be the column of the current compensated second-layer weight matrix $\widetilde{W}_2^t$ associated with neuron i. Then the cost of removing i is exactly:

Proposition 2 (Exact error increase after one removal)**.** *For every $i \in K^t$,*

$$\varepsilon\big(K^t \setminus \{i\}\big) - \varepsilon(K^t) = \frac{\|w_i^t\|_2^2}{\Omega_{ii}^t}. \tag{14}$$

The proof is given in Appendix D. The score therefore combines the part of the neuron that cannot be recovered, $1/\Omega_{ii}^t$, with its effect on the MLP output, $\|w_i^t\|_2^2$. Finally, the neuron removed at step t is

$$j_t = \arg\min_{i \in K^t} \frac{\|w_i^t\|_2^2}{\Omega_{ii}^t}. \tag{15}$$

Unlike a fixed importance score, this cost changes after every removal, as each neuron is predicted from a different set of survivors.

Efficient updates. After removing j_t, its contribution is compensated for by the remaining neurons. With $K^{t+1} = K^t \setminus \{j_t\}$, the compensated weights and precision matrix are updated in closed form:

$$w_i^{t+1} = w_i^t - \frac{\Omega_{ij_t}^t}{\Omega_{j_t j_t}^t} w_{j_t}^t, \qquad i \in K^{t+1}, \tag{16}$$

$$\Omega^{t+1} = \Omega_{K^{t+1}K^{t+1}}^t - \frac{\Omega_{K^{t+1}j_t}^t \Omega_{j_t K^{t+1}}^t}{\Omega_{j_t j_t}^t}. \tag{17}$$

Thus, no covariance needs to be recomputed and no new matrix inversion is required. The resulting nested elimination order can be reused for all target sparsities.

3.2 Global allocation across Transformer blocks

The previous section identifies the locally cheapest neuron to remove within each block. A second question remains: *from which block should the next neuron be removed?* CORP and GRAIL use the same sparsity across blocks (Zhang & Yang, 2026; Tang et al., 2026), whereas VBP and DVBP allow non-uniform allocation through global ranking (Berisha et al., 2025; Lee et al., 2026). However, their global decisions rely on raw neuron scores rather than on the post-compensation error derived in Section 3.1. More generally, raw scores are not directly comparable across Transformer blocks because their activation and output scales can differ significantly.

We therefore build the global allocation directly from the local pruning cost of Proposition 2. Restoring the block index, let K_b^t be the surviving neurons in block b after t local removals, and let $j_{b,t}$ be the next neuron selected by equation 15. Its exact local cost is

$$c_{b,t} := \varepsilon_b\big(K_b^t \setminus \{j_{b,t}\}\big) - \varepsilon_b(K_b^t) = \frac{\|w_{b,j_{b,t}}^t\|_2^2}{\Omega_{b,j_{b,t}j_{b,t}}^t}. \tag{18}$$

Making the error relative. The MLP output is added to the residual stream, so the same absolute reconstruction error can have a different importance depending on the scale of the signal at that block. Let $r^b \in \mathbb{R}^{D^b}$ denote the residual-stream representation entering the whole Transformer block b, before its attention and MLP sublayers. We define

$$V_b = \mathrm{Tr}\left(\mathrm{Cov}(r^b)\right). \tag{19}$$

Dividing by V_b expresses the pruning error relative to the signal present at block b.

We therefore define the global score of the next removal in block b as

$$s_{b,t} = \frac{c_{b,t}}{V_b} = \frac{\varepsilon_b(K_b^t \setminus \{j_{b,t}\}) - \varepsilon_b(K_b^t)}{V_b}. \tag{20}$$

The global allocator compares $s_{b,t}$ across blocks and removes the available candidate with the smallest score. Since V_b is constant within a block, this normalisation does not change the local elimination order; it only makes the costs comparable across blocks. The allocation ablations are reported in Section 4.3 and Appendix I.

3.3 Statistics and algorithm

CARVE uses a small calibration set to estimate, for each block b, the hidden mean and covariance (μ_b, Σ_b) and the residual-stream variance V_b. All pruning scores are then computed in closed form from these statistics and the model weights. Following numerically stable estimators (Welford, 1962; Chan et al., 1983), we accumulate centred second moments directly rather than computing $\mathbb{E}[h^b {h^b}^\top] - \mu_b \mu_b^\top$. Since the empirical covariance can be singular, we initialise the precision matrix with a small ridge:

$$\Omega_b^0 = \big(\Sigma_b + \lambda_b I\big)^{-1}, \qquad \lambda_b = \rho\, \overline{\mathrm{diag}(\Sigma_b)}, \tag{21}$$

where $\rho = 10^{-3}$ is fixed for all blocks and models. The previous derivations correspond to the unregularised case, while equation 21 is used in practice. Appendix G verifies these closed-form expressions numerically, and Appendix H studies the sensitivity to ρ.

For each block, CARVE computes the nested elimination order of Section 3.1 once. The global allocator then repeatedly selects the block with the smallest score $s_{b,t}$ until the target pruning budget is reached. Finally, the optimal affine compensation of equation 9 is applied to the final kept set of each block. No gradients, fine-tuning, distillation, or post-pruning recovery are used. The complete calibration and pruning procedures are summarised in Algorithms B.1 and B.2 in Appendix B.

4 EXPERIMENTS

In our experiments, we show that CARVE consistently improves performance across model scales, architecture families, and datasets. We isolate the effects of neuron selection and cross-block allocation, and show that allocation becomes critical under aggressive pruning.

Table 1: **ImageNet-1k Top-1 accuracy (%) [↑] at 50% MLP sparsity.** All methods are evaluated in the post-training setting without fine-tuning. Best result is in **bold** and second best is underlined. Teacher , competing pruning methods , and our method .

Model	Teacher	Baselines						Ours
	Dense	MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
Vision Transformers								
DeiT-Tiny	72.0	0.5	12.0	42.2	55.9	55.3	56.6	**59.7**
DeiT-Small	79.7	0.4	39.9	65.7	69.7	69.2	70.2	**72.9**
DeiT-Base	81.7	0.2	47.1	68.2	75.2	72.3	73.2	**77.3**
ViT-B/16	81.1	0.2	36.4	59.9	72.8	73.8	74.0	**76.4**
DeiT-III-Large	84.6	0.7	70.9	76.8	78.9	82.0	82.2	**83.0**
DeiT-III-Huge	85.0	0.6	78.7	81.1	82.5	84.1	84.2	**84.5**
ConvNeXt								
ConvNeXt-T	84.0	0.1	1.0	14.0	35.0	60.5	62.3	**67.6**
ConvNeXt-S	84.9	0.1	0.8	27.0	40.7	63.3	64.7	**73.7**
ConvNeXt-B	85.5	0.1	2.9	56.5	64.5	75.4	75.8	**76.9**

4.1 SETUP

Models and data. Our vision backbones include DeiT-Tiny, DeiT-Small, and DeiT-Base (Touvron et al., 2021), ViT-B/16 (Dosovitskiy et al., 2021), DeiT-III-Large and DeiT-III-Huge (Touvron et al., 2022), and ConvNeXt-T/S/B (Liu et al., 2022). They span 5.7M–632M parameters and 12–36 blocks. Our main vision benchmark is ImageNet-1k (Russakovsky et al., 2015), evaluated on the full 50,000-image validation set. We also evaluate the same backbones on CIFAR-100 (Krizhevsky, 2009) after fine-tuning them on this dataset. For language modelling, we evaluate Qwen2.5-1.5B (Qwen et al., 2025) on WikiText-2 (Merity et al., 2017).

Baselines. We compare CARVE against two classical uncompensated pruning criteria, group magnitude (MAG) (Han et al., 2015) and SNIP (Lee et al., 2019), as well as compensated post-training methods including VBP (Berisha et al., 2025). We also evaluate recent concurrent methods: DVBP (Lee et al., 2026), CORP (Zhang & Yang, 2026), and GRAIL (Tang et al., 2026). GRAIL is selector-agnostic and uses the Wanda score adopted in its vision experiments. For CORP, we implement only the MLP branch, leaving attention-head pruning out of scope.

Protocol. For a fair comparison, all methods are either reimplemented or adapted from the authors' code within the same codebase and evaluated under a common protocol. Compression is performed in a one-shot, post-training setting and affects only the MLP hidden width. Results are reported without post-pruning fine-tuning, distillation, or optimisation-based recovery, so that performance reflects the pruning method itself. Sparsity denotes the fraction of MLP hidden neurons removed and ranges from 10% to 90%. At a given sparsity, all methods use the same pruning budget; only neuron selection, compensation, and cross-block allocation differ. All methods use the same class-balanced calibration subset of 5,000 training images. Appendix Q evaluates sensitivity to the choice

Table 2: **Accuracy-gap recovery by CARVE (%).** Fraction of each baseline-to-teacher gap recovered by CARVE.

	VBP	DVBP	CORP	GRAIL
All (10–90%)	47.1	31.7	21.0	**20.2**
$\leq 70\%$	63.5	47.0	28.9	**26.2**

Table 3: **Accuracy–efficiency trade-off on DeiT-Base.**

MLP sparsity	Params (M)	MACs (G)	CARVE Top-1
0%	86.6	17.58	81.7
10%	80.9 (−6.5%)	16.47 (−6.3%)	81.7
30%	69.6 (−19.6%)	14.24 (−19.0%)	81.1
50%	58.2 (−32.7%)	12.01 (−31.7%)	77.3
70%	46.9 (−45.8%)	9.77 (−44.4%)	69.1

of calibration samples. CARVE uses $\rho = 10^{-3}$ for all models and sparsities. Baselines use the hyperparameters recommended by their authors, with the same calibration-set size across methods. Results are averaged over three calibration seeds, and each experiment is run on a single NVIDIA B200 GPU.

4.2 Main results

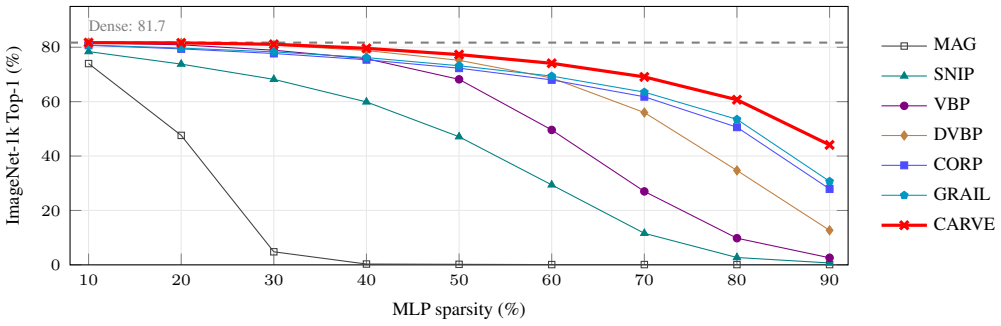

Figure 1: ImageNet-1k Top-1 accuracy at increasing sparsity on DeiT-Base.

Quantitative results on ImageNet-1k. Table 1 reports Top-1 accuracy at 50% MLP sparsity across the nine vision backbones. CARVE achieves the highest accuracy on every model, closely followed by GRAIL. Its advantage is particularly pronounced on smaller and medium-scale backbones, reaching $+9.0$ points on ConvNeXt-S, $+3.1$ on DeiT-Tiny, and $+2.4$ on ViT-B/16 over the strongest competitor. The gap narrows on larger models, down to $+0.3$ points on DeiT-III-Huge, where CARVE remains only 0.5 points below the dense teacher despite removing half of the MLP neurons. Figure 1 complements this fixed-sparsity comparison by showing performance across the full range of sparsity levels on DeiT-Base. Considering the main configuration of each method, CARVE matches or outperforms all competing methods across the full sparsity range, with larger gains as sparsity increases. It matches DVBP at 10–20% sparsity, then leads the strongest competitor by 2.1 points at 50%, 5.6 at 70%, and 13.4 at 90%. Note that in these experiments, methods use their respective cross-block allocation strategies; we isolate this factor in Section 4.3.

Absolute gains naturally shrink when a baseline is already close to the dense teacher. We therefore also report the proportion of its remaining teacher gap recovered by CARVE: for a baseline with accuracy a_{base} and teacher accuracy a_T, we define $\text{Recovery} = \frac{a_{\text{CARVE}} - a_{\text{base}}}{a_T - a_{\text{base}}}$. Table 2 shows that CARVE closes 31.7% of DVBP's remaining teacher gap and 20.2% of GRAIL's across all sparsities. Up to 70% sparsity, these fractions rise to 47.0% and 26.2%, respectively.

Additional evaluations. The complete ImageNet-1k results across all backbones, sparsity levels, and calibration seeds are reported in Appendix J, with the full gap-recovery analysis in Appendix K. The same trends hold on CIFAR-100 (Appendix P). As an extension beyond vision, we also apply CARVE to Qwen2.5-1.5B on WikiText-2, showing that the same pruning principle extends to a decoder-only language model. The results are also robust to the two choices of the protocol: Top-1 changes by at most 0.4 points for ρ between 10^{-6} and 10^{-2} (Appendix H), and the margin of CARVE does not depend on the size of the calibration set once it is representative enough (Appendix Q). Finally, the closed-form quantities of Section 3.1 match their measured values to machine precision, or up to the known effect of the ridge (Appendix G).

Efficiency and compression. CARVE requires one forward pass to collect activation statistics, followed by greedy elimination and closed-form compensation. Elimination is performed independently across blocks and in parallel on the GPU, and its cost is independent of the target sparsity. On a single NVIDIA B200, the full compression takes 30 s for DeiT-Base, 2.9 min for DeiT-III-Huge, and 3.5 min for Qwen2.5-1.5B.

For fixed-width Transformers, a given neuron budget yields the same parameter and MAC reduction regardless of its allocation across blocks. Table 3 shows this trade-off on DeiT-Base. Since only the MLP hidden width is pruned, the overall model reduction is smaller than the nominal MLP sparsity: at 50% MLP sparsity, parameters and MACs decrease by 32.7% and 31.7%, respectively, while CARVE reaches 77.3% Top-1 accuracy. The MLP accounts for 65.5% of the dense model's parameters, while attention, embeddings, and the classifier remain unchanged. Appendix O reports parameters, MACs and peak memory for every backbone; on ConvNeXt, whose stages have different widths, the same sparsity gives different MACs and memory depending on the allocation.

4.3 Disentangling selection and allocation

CARVE differs from existing methods in both its local pruning rule and its cross-block allocation. We isolate these two effects below.

Selection under fixed allocation. To isolate the effects of neuron selection and compensation, we introduce VBP_L and CARVE_L, which use a uniform sparsity across all blocks. Table 4 shows that, at 50% MLP sparsity, CARVE_L achieves the highest accuracy on eight of the nine backbones. The gain over the strongest baseline remains modest on the largest Transformers, but reaches $+5.6$ and $+8.0$ points on ConvNeXt-T and ConvNeXt-S, respectively. The full sparsity evaluation in Appendix L confirms this trend: CARVE_L wins 78 of the 81 model-sparsity settings, with the only exceptions occurring on DeiT-III-Huge at 50% and 60% sparsity and on ConvNeXt-B at 80%. Across the full grid, it improves over the strongest uniform-per-block-sparsity baseline by $+2.5$ Top-1 points on average, showing that selection alone accounts for a substantial part of the gain.

Table 4: **Top-1 accuracy (%) [↑] at 50% MLP sparsity with a uniform per-block sparsity.** This setting isolates the neuron selection rule from the global allocation. The best result is in **bold** and the second-best is underlined. Teacher , competing pruning methods , and our method .

Model	Teacher	Baselines						Ours
	Dense	MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
Vision Transformers								
DeiT-Tiny	72.0	0.5	12.0	30.5	51.9	55.3	56.6	**58.0**
DeiT-Small	79.7	0.4	39.9	57.2	67.1	69.2	70.2	**71.2**
DeiT-Base	81.7	0.2	47.1	63.4	70.1	72.3	73.2	**74.3**
ViT-B/16	81.1	0.2	36.4	65.3	71.8	73.8	74.0	**74.7**
DeiT-III-Large	84.6	0.7	70.9	79.0	81.0	82.0	82.2	**82.5**
DeiT-III-Huge	85.0	0.6	78.7	82.5	83.7	84.1	**84.2**	83.9
ConvNeXt								
ConvNeXt-T	84.0	0.1	1.0	32.2	57.4	60.5	62.3	**67.9**
ConvNeXt-S	84.9	0.1	0.8	31.8	62.7	63.3	64.7	**72.7**
ConvNeXt-B	85.5	0.1	2.9	57.8	74.7	75.4	75.8	**77.8**

Cross-block allocation. We next investigate why local pruning costs are not directly comparable across blocks. To assess the role of cross-block normalisation, we introduce CARVE_G, which directly compares raw CARVE scores across blocks, without cross-block normalisation. Figure 2(a) shows that raw scores can produce highly unbalanced allocations: at 70% sparsity on DeiT-Base, VBP, DVBP, and CARVE_G heavily prune early blocks while preserving much larger fractions of later ones. Appendix I extends this comparison to six backbones and all sparsities.

The reason is that local reconstruction errors do not account for errors already accumulated in the residual stream. The local objective in equation 6 is defined on the teacher input, so a block may

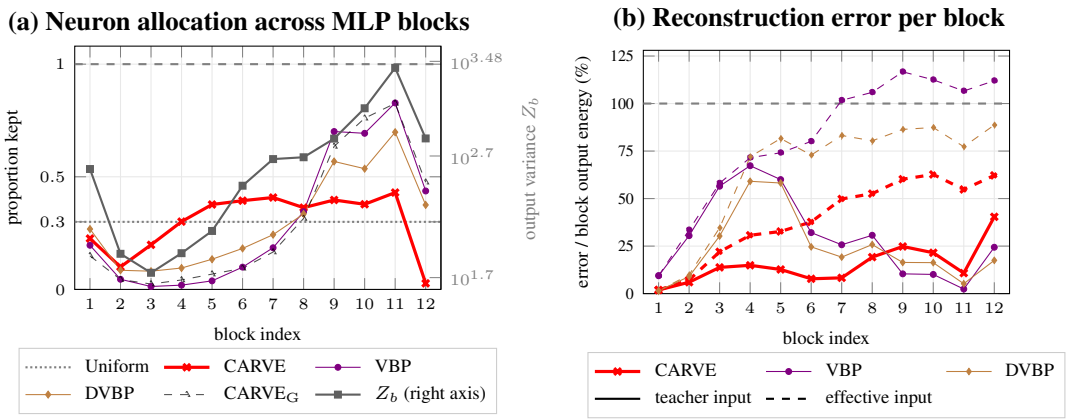

Figure 2: Cross-block allocation on DeiT-Base (ImageNet-1k) at 70% MLP sparsity. **(a)** Fraction of neurons retained per block under a global pruning budget, with dense MLP output variance Z_b on the right axis. **(b)** Reconstruction error relative to block output energy, evaluated on teacher inputs (solid) and effective inputs from the pruned network (dashed).

have a small reconstruction error on the clean input but a much larger one on the input actually produced by the pruned network, with the gap increasing with depth. Figure 2(b) shows both cases, using solid curves for teacher inputs and dashed curves for effective inputs. The latter reflects the error actually encountered by the pruned network and reveals how upstream corruption can turn an apparently good local allocation into a poor global one. For VBP, late blocks appear well preserved on clean inputs: block 11 loses only 2% of its output energy, yet its error exceeds 100% on the effective input, i.e., it becomes larger than the signal being reconstructed. Averaged over blocks 9 to 12, 90% of VBP's reconstruction error comes from errors propagated by earlier blocks rather than from the local pruning itself (84% for DVBP and 92% for CARVE$_G$, versus 60% for CARVE, which keeps fewer neurons in these later blocks). This cross-block mismatch becomes even more pronounced on the 32-block DeiT-III-Huge: raw scores differ by two orders of magnitude or more across depth, and comparing them directly can drive some blocks to 99% sparsity. See Appendix N.

5 Conclusion

We introduced CARVE, a post-training structured pruning method that selects MLP neurons according to the error that remains after optimal compensation, rather than according to a fixed pre-compensation importance score. This leads to a dynamic greedy criterion that adapts as the retained set evolves and can be evaluated efficiently through closed-form rank-one updates. We further showed that cross-block allocation becomes critical at high sparsity and introduced a residual-aware strategy for comparing pruning costs across blocks. Across nine vision backbones, CARVE consistently improves over strong post-training pruning baselines, with particularly large gains under aggressive compression. More broadly, CARVE points toward structured pruning strategies that jointly account for local recoverability and the global distribution of pruning errors across blocks, providing a broader basis for compression across architectures and sparsity regimes.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the manuscript, including suggestions for clarity, structure, and reformulation, and as a coding assistant. Generative AI tools were not used to develop the proposed method, formulate the initial research idea, write the initial draft of the paper, or generate experimental results. All AI-assisted text and code were reviewed by the authors, and AI-assisted code was verified and tested before use. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10865–10873, 2024. doi: 10.1609/aaai.v38i10.28960. URL https://doi.org/10.1609/aaai.v38i10.28960.

Uranik Berisha, Jens Mehnert, and Alexandru Paul Condurache. Variance-based pruning for accelerating and compressing trained networks. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Tony F. Chan, Gene H. Golub, and Randall J. Leveque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983. doi: 10.1080/00031305.1983.10483115. URL https://doi.org/10.1080/00031305.1983.10483115.

Christophe Couvreur and Yoram Bresler. On the optimality of the backward greedy algorithm for the subset selection problem. *SIAM Journal on Matrix Analysis and Applications*, 21(3):797–808, 2000. ISSN 0895-4798. doi: 10.1137/S0895479898332928. Copyright: Copyright 2017 Elsevier B.V., All rights reserved.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In Lise Getoor and Tobias Scheffer (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 1057–1064. Omnipress, 2011. URL https://icml.cc/2011/papers/542_icmlpaper.pdf.

Amit Deshpande, Luis Rademacher, Santosh S. Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2(12):225–247, 2006. doi: 10.4086/toc.2006.v002a012. URL https://theoryofcomputing.org/articles/v002a012.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ethan R. Elenberg, Rajiv Khanna, Alexandros G. Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539 – 3568, 2018. doi: 10.1214/17-AOS1679. URL https://doi.org/10.1214/17-AOS1679.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4475–4488. Curran Associates, Inc., 2022. doi: 10.52202/068431-0323. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1caf09c9f4e6b0150b06a07e77f2710c-Paper-Conference.pdf.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 28, pp. 1135–1143, 2015.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles (eds.), *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Geon Tack Lee, Jaegul Choo, and Kang Eun Jeon. Denoised variance-based pruning with optimal brain bias compensation. In *European Conference on Computer Vision (ECCV)*, 2026.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. In *ICLR*, 2019.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJqFGTslg.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, June 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

Chengchao Shen, Hourun Zhu, Gongfan Fang, Jianxin Wang, and Xinchao Wang. Diversity-guided mlp reduction for efficient large vision transformers. *arXiv preprint arXiv:2506.08591*, 2025.

Yaroslav Shitov. Column subset selection is np-complete. *Linear Algebra and its Applications*, 610: 52–58, 2021. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2020.09.015. URL https://www.sciencedirect.com/science/article/pii/S0024379520304377.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/14c856c7a41297804de4c4890e846b25-Paper-Conference.pdf.

Wenwu Tang, Dong Wang, Lothar Thiele, and Olga Saukh. Grail: Post-hoc compensation by linear reconstruction for compressed networks. In Rebekka Burkholz, Shiwei Liu, Saiprasad Ravishankar, William Redman, Wei Huang, Weijie Su, and Zhihui Zhu (eds.), *Conference on Parsimony and Learning*, volume 328 of *Proceedings of Machine Learning Research*, pp. 881–895. PMLR, 23–26 Mar 2026. URL https://proceedings.mlr.press/v328/tang26a.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 516–533, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20053-3.

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4:419–420, 1962. URL https://api.semanticscholar.org/CorpusID:120126049.

Boxiang Zhang and Baijian Yang. Corp: Closed-form one-shot representation-preserving structured pruning for transformers, 2026. URL https://arxiv.org/abs/2602.05243.

Supplementary: Prune What Can Be Reconstructed: CARVE for Compensation-Aware Neuron Selection

TABLE OF CONTENTS: SUPPLEMENTARY MATERIAL

I Theoretical results **15**

A Notation **15**

B Algorithms **16**

C Optimal affine compensation **17**

D Exact updates **19**

E Architectural variants **20**

F Column subset selection **21**

II Additional experiments **22**

G Numerical validation **22**

H Ridge sensitivity **24**

I Ablation: normalisation **25**

J ImageNet-1k results **26**

K Gap closure **30**

L Uniform per-block sparsity results **30**

M Language modelling **34**

N Allocation on a deeper Transformer **34**

O Computational cost **35**

P Results on the CIFAR-100 dataset **37**

Q Calibration set **42**

I Theoretical results

A NOTATION

Every symbol used in the paper is listed below. The superscript b indexes the Transformer block and is dropped whenever a statement concerns a single block. Matrix subscripts select a submatrix by index set, so Σ_{KP} has its rows in K and its columns in P, and $W_{2,K}$ is the columns of W_2 indexed by K. Vectors are columns, and every expectation is taken over the calibration distribution. Orthogonality of square-integrable random vectors is with respect to the inner product $\langle u, v\rangle = \mathbb{E}[u^\top v]$.

Symbol	Meaning	Def.
Indices, dimensions and counts		
b	Transformer block index, $1 \le b \le B$	§3.1
B	number of Transformer blocks	§3.1
D, D^b	embedding, i.e. residual-stream, width	§3.1
H, H^b	MLP hidden width; the pruned axis	§3.1
i, j	hidden-neuron indices	§3.1
d	output-coordinate index	(D.53)
t	greedy elimination step	(15)
k	number of kept neurons, $k = \lvert K\rvert$	(6)
N	calibration rows (tokens or patches)	App. F
$\mathcal{D}$	unlabelled calibration set	Alg. B.1
ρ	relative ridge, fixed to 10^{-3}	(21)
The dense block		
x^b	MLP input, after the normalisation	(1)
r^b	residual stream entering block b	(19)
ϕ	elementwise activation (GELU here)	(1)
h^b	hidden rep. $\phi(W_1^b x^b + c_1^b)$	(1)
y^b	MLP output $W_2^b h^b + c_2^b$	(1)
W_1^b, c_1^b	first linear layer	(1)
W_2^b, c_2^b	second linear layer	(1)
U, c_u	second input projection, gated MLP	(E.62)
g	per-channel gain (LayerScale)	(E.56)
z^b	$g \odot y^b$, what reaches the stream	(E.56)
$\widehat{W}_2$	effective weights $\mathrm{diag}(g)W_2$	(E.59)
Sets and partitions		
K, $P = K^c$	kept and removed neurons	(3)
K^t	survivors after t removals	(15)
K^{t+1}	$K^t \setminus \{j_t\}$	(D.47)
K_b^t	survivors in block b	(18)
$K^0, \dots, K^m$	an elimination chain down to K	Cor. 3
j_t, $j_{b,t}$	neuron removed at step t	(15)
h_K, h_P	kept and removed parts of h	(3)
$W_{2,K}$, $W_{2,P}$	column blocks of W_2	(3)
$\mathcal{A}_J$	affine span of h_J, closed in L^2	(D.52)
$\Pi_{\mathcal{A}_K}$	orthogonal projection onto $\mathcal{A}_K$	(D.53)
Calibration statistics		
μ_b, Σ_b	hidden mean and covariance	§3.1
Σ_{KK}, Σ_{KP}	blocks of Σ under $K \sqcup P$	(7)
Ω^t	precision matrix $\Sigma_{K^tK^t}^{-1}$	§3.1
$1/\Omega_{ii}^t$	residual variance of neuron i	(13)
$\Sigma_{P\mid K}^{\mathrm{res}}$	unrecoverable part of h_P given h_K	(10)
$\sigma_{j_t}^2$	Schur complement; $= 1/\Omega_{j_tj_t}^t$	(D.48)
S, u, v	blocks of $\Sigma_{K^tK^t}$ at j_t	(D.47)
V_b	variance already in the residual stream	(19)
Z_b	block's own MLP output variance	Fig. 2
λ_b	absolute ridge $\rho\,\overline{\mathrm{diag}(\Sigma_b)}$	(21)
Ω_b^0	$(\Sigma_b + \lambda_b I)^{-1}$, the starting state	(21)
Welford vars.	$\mathcal{Z}, n, \mu, M, q, \bar{z}, C, \delta$; tag $\epsilon \in \{h, r\}$	Alg. B.1

Symbol	Meaning	Def.
Compensation and its error		
$\widetilde{W}_2$, $\widetilde{c}_2$	compensated second layer	(5)
$\widetilde{y}$	output of the compressed block	(5)
β	$\Sigma_{KK}^{-1}\Sigma_{KP}$, predicts h_P from h_K	(8)
α	the same regression, one-neuron case	(13)
$\beta^{(j_t)}$	one-neuron case, read off Ω^t	(D.49)
η_P	affine prediction residual of h_P	(C.29)
$\varepsilon(K)$, $\varepsilon_b(K)$	error after optimal compensation	(10)
$\varepsilon_k^\star$	its minimum over all $\lvert K\rvert = k$	(6)
Elimination and allocation		
w_i^t	column of $\widetilde{W}_2^t$ for neuron i	§3.1
$\lVert w_i^t\rVert_2^2/\Omega_{ii}^t$	exact cost of removing i	(14)
$s_i(K)$	that cost evaluated at kept set K	(D.55)
δ_t	difference of projections at step t; its squared norm is the error increment	App. D
Δ_i^b	cost charged when i is removed	Alg. B.2
$c_{b,t}$	local cost of the next removal in b	(18)
$s_{b,t}$	$c_{b,t}/V_b$, comparable across blocks	(20)
s_i^b	Δ_i^b/V_b, compared by the allocator	Alg. B.2
κ	target sparsity: fraction of neurons removed	Alg. B.2
Local to Appendix F		
A, A_K	centred activations; its kept columns	(F.64)
X, X_K	classical CSSP instance and its kept columns	(F.65)
Θ, n_0	its coefficient matrix and row count	(F.65)
Operators		
$\mathbb{E}$, Var, Cov	over the calibration distribution	
Tr, diag	trace; diagonal matrix from a vector	
$\overline{\mathrm{diag}(\Sigma)}$	mean of the diagonal of Σ	
$\odot$	elementwise product	
$\lVert\cdot\rVert_2$, $\lVert\cdot\rVert_F$	Euclidean and Frobenius norms	
$\lVert\cdot\rVert_{L^2}$	L^2 norm of a random variable	
$(\cdot)^\top$, $(\cdot)^\dagger$	transpose; pseudoinverse	
$\sqcup, \lvert\cdot\rvert, \setminus$	disjoint union, cardinality, difference	
$\succ 0$, $\succeq 0$	positive definite, positive semi-definite	
$\mathbf{1}$, I, I_N	ones vector; identity, of size N	
Method names		
CARVE	selection of §3.1 + allocation of §3.2	
$\mathrm{CARVE}_{\mathrm{L}}$	the same at a uniform per-block sparsity	
$\mathrm{CARVE}_{\mathrm{G}}$	raw costs compared directly, no normalisation	
$\mathrm{VBP}_{\mathrm{L}}$	uniform per-block sparsity	
$\mathrm{DVBP}_{\mathrm{L}}$	DVBP at a uniform per-block sparsity	
Baselines	uncompensated: MAG, SNIP, Wanda; compensated: VBP, DVBP, CORP, GRAIL	

B ALGORITHMS

This appendix gives the pseudocode of the calibration pass (Algorithm B.1) and of CARVE pruning (Algorithm B.2). Algorithm B.1 performs the calibration of Section 3.3: one pass over the calibration set accumulates, per block and in a numerically stable way, the mean and centred second moment of the post-activation hidden representation and of the residual stream. Algorithm B.2 performs CARVE pruning: the per-block greedy elimination, which is independent across blocks and therefore parallel, then the global allocation over the normalised costs, and finally the affine compensation of each block's kept set.

Algorithm B.1 Statistics estimation

Require: unlabelled calibration set $\mathcal{D}$

Initialise $n_b^\epsilon, \mu_b^\epsilon, M_b^\epsilon \leftarrow 0$ for every block b and $\epsilon \in \{h, r\}$
for each calibration batch **do**
  forward the batch and collect flattened rows $\mathcal{Z}_b^h$ of h^b and $\mathcal{Z}_b^r$ of r^b
  for each block b **do**
    for $\epsilon \in \{h, r\}$ **do**
      $q_b^\epsilon \leftarrow |\mathcal{Z}_b^\epsilon|;\ \bar{z}_b^\epsilon \leftarrow (q_b^\epsilon)^{-1} \sum_{z \in \mathcal{Z}_b^\epsilon} z$
      $C_b^\epsilon \leftarrow \sum_{z \in \mathcal{Z}_b^\epsilon} (z - \bar{z}_b^\epsilon)(z - \bar{z}_b^\epsilon)^\top$
      $\delta_b^\epsilon \leftarrow \bar{z}_b^\epsilon - \mu_b^\epsilon$
      $\mu_b^\epsilon \leftarrow \mu_b^\epsilon + \frac{q_b^\epsilon}{n_b^\epsilon + q_b^\epsilon} \delta_b^\epsilon$
      $M_b^\epsilon \leftarrow M_b^\epsilon + C_b^\epsilon + \frac{n_b^\epsilon q_b^\epsilon}{n_b^\epsilon + q_b^\epsilon} \delta_b^\epsilon \delta_b^{\epsilon\top}$
      $n_b^\epsilon \leftarrow n_b^\epsilon + q_b^\epsilon$
    end for
  end for
end for
$\Sigma_b \leftarrow M_b^h / n_b^h; \quad V_b \leftarrow \mathrm{Tr}(M_b^r / n_b^r)$
return $\{\mu_b^h, \Sigma_b, V_b\}_{b=1}^B$

Algorithm B.2 CARVE

Require: per block b: μ_b, Σ_b, V_b, W_2^b, c_2^b; ridge ρ; target sparsity κ

for all blocks b in parallel **do**
  initialise $K_b^0 \leftarrow \{1, \ldots, H^b\}$, $\Omega^0 \leftarrow (\Sigma_b + \lambda_b I)^{-1}$ as in equation 21, and $w_i^0 \leftarrow W_2^b[:, i]$
  for $t = 0, \ldots, H^b - 2$ **do**
    $j_{b,t} \leftarrow \arg\min_{i \in K_b^t} \|w_i^t\|_2^2 / \Omega_{ii}^t$
    $\Delta_{j_{b,t}}^b \leftarrow \|w_{j_{b,t}}^t\|_2^2 / \Omega_{j_{b,t} j_{b,t}}^t$
    update K_b^t, w^t, Ω^t to step $t+1$ using equation 16 and equation 17
  end for
  let i be the one neuron left in $K_b^{H^b - 1}$; set $\Delta_i^b \leftarrow \|w_i\|_2^2 / \Omega_{ii}$ on that final state
end for
for each neuron i in every block b, set $s_i^b \leftarrow \Delta_i^b / V_b$
initialise a global heap with the first neuron $j_{b,0}$ of each block's elimination order
while $\sum_b |K_b| > \mathrm{round}\big((1 - \kappa) \sum_b H^b\big)$ **do**
  pop the neuron with smallest s_i^b and prune it; insert the next neuron in block b into the heap
end while
for all blocks b **do**
  apply the affine compensation equation 9 to the final kept set K_b
end for
return compressed MLPs with the selected widths and compensated weights and biases

C Optimal affine compensation

We prove Proposition 1 for one block and omit the block index. Let $\{1, \dots, H\} = K \sqcup P$, with $\mathbb{E}\|h\|_2^2 < \infty$ and $\Sigma_{KK} \succ 0$. As in Section 3.1, we partition both the hidden representation and the corresponding columns of the second MLP layer:

$$h = \begin{bmatrix} h_K \\ h_P \end{bmatrix}, \qquad W_2 = [W_{2,K} \quad W_{2,P}]. \tag{C.22}$$

The original and compressed outputs are therefore

$$y = W_{2,K} h_K + W_{2,P} h_P + c_2, \qquad \widetilde{y} = \widetilde{W}_2 h_K + \widetilde{c}_2. \tag{C.23}$$

For a fixed kept set K, the objective is

$$\min_{\widetilde{W}_2, \widetilde{c}_2} \mathbb{E}\left[\|y - \widetilde{y}\|_2^2\right]. \tag{C.24}$$

Optimal affine predictor. For a fixed kept set K, we first ask how well the removed activations h_P can be reconstructed from the surviving activations h_K. We consider the affine predictor

$$\widehat{h}_P = \mu_P + \beta^\top (h_K - \mu_K). \tag{C.25}$$

and choose β to minimise the mean squared prediction error:

$$\min_{\beta} \mathbb{E}\left[\left\|h_P - \mu_P - \beta^\top (h_K - \mu_K)\right\|_2^2\right]. \tag{C.26}$$

Setting the derivative with respect to β to zero gives the normal equation

$$\Sigma_{KK}\beta = \Sigma_{KP}, \tag{C.27}$$

and therefore

$$\beta = \Sigma_{KK}^{-1}\Sigma_{KP}. \tag{C.28}$$

We can then write the removed representation as its optimal affine prediction from h_K plus the corresponding residual:

$$h_P = \underbrace{\mu_P + \beta^\top (h_K - \mu_K)}_{\text{predicted from } h_K} + \underbrace{\left[h_P - \mu_P - \beta^\top (h_K - \mu_K)\right]}_{\eta_P}. \tag{C.29}$$

The normal equation gives

$$\begin{aligned} \mathbb{E}[\eta_P] &= 0, \\ \mathbb{E}\left[\eta_P (h_K - \mu_K)^\top\right] &= \Sigma_{PK} - \beta^\top \Sigma_{KK} = 0. \end{aligned} \tag{C.30}$$

Thus the residual has zero mean and is uncorrelated with the kept activations.

Substituting equation C.29 into the original output (equation C.23) gives

$$\begin{aligned} y = W_{2,K} h_K + W_{2,P}\left[\mu_P + \beta^\top (h_K - \mu_K)\right] + c_2 \\ + W_{2,P}\left[h_P - \mu_P - \beta^\top (h_K - \mu_K)\right]. \end{aligned} \tag{C.31}$$

Collecting the terms that depend on h_K gives

$$\begin{aligned} y = \ & \left(W_{2,K} + W_{2,P}\beta^\top\right) h_K \\ & + c_2 + W_{2,P}\mu_P - W_{2,P}\beta^\top \mu_K \\ & + W_{2,P}\left[h_P - \mu_P - \beta^\top (h_K - \mu_K)\right]. \end{aligned} \tag{C.32}$$

The first two lines are affine functions of the surviving representation h_K and can therefore be represented exactly by the compressed MLP. The last line is orthogonal to every affine function of h_K by Eq. equation C.30. No further affine compensation using h_K can therefore reduce the expected squared output error. Hence,

$$\widetilde{W}_2^\star = W_{2,K} + W_{2,P}\beta^\top, \qquad \widetilde{c}_2^\star = c_2 + W_{2,P}\mu_P - W_{2,P}\beta^\top \mu_K, \tag{C.33}$$

which proves Proposition 1.

Since $\mathbb{E}[\eta_P] = 0$, taking expectations in equation C.32 also gives

$$\mathbb{E}[\widetilde{y}] = \mathbb{E}[y], \tag{C.34}$$

so the compensation preserves the output mean exactly.

Residual reconstruction error. After absorbing the predictable part into the surviving weights, the only difference between the original and compensated outputs is therefore

$$y - \widetilde{y} = W_{2,P}\eta_P. \tag{C.35}$$

We now compute the covariance of η_P. Since both $h_P - \mu_P$ and $h_K - \mu_K$ are centred,

$$\begin{aligned}\mathrm{Cov}(\eta_P) &= \mathbb{E}\left[\big(h_P - \mu_P - \beta^\top(h_K - \mu_K)\big)\big(h_P - \mu_P - \beta^\top(h_K - \mu_K)\big)^\top\right] \\ &= \Sigma_{PP} - \Sigma_{PK}\beta - \beta^\top\Sigma_{KP} + \beta^\top\Sigma_{KK}\beta. \end{aligned} \tag{C.36}$$

Using the optimal regression coefficient

$$\beta = \Sigma_{KK}^{-1}\Sigma_{KP},$$

the last three terms simplify, giving

$$\mathrm{Cov}(\eta_P) = \Sigma_{PP} - \Sigma_{PK}\Sigma_{KK}^{-1}\Sigma_{KP} = \Sigma_{P|K}^{\mathrm{res}}. \tag{C.37}$$

Thus, $\Sigma_{P|K}^{\mathrm{res}}$ is exactly the covariance of the residual after the optimal affine prediction of h_P from h_K.

Finally, since η_P has zero mean and $y - \widetilde{y} = W_{2,P}\eta_P$,

$$\begin{aligned}\varepsilon(K) &= \mathbb{E}\left[\|y - \widetilde{y}\|_2^2\right] \\ &= \mathbb{E}\left[\|W_{2,P}\eta_P\|_2^2\right] \\ &= \mathrm{Tr}\left(W_{2,P}\mathbb{E}[\eta_P\eta_P^\top]W_{2,P}^\top\right). \end{aligned} \tag{C.38}$$

Because $\mathbb{E}[\eta_P\eta_P^\top] = \Sigma_{P|K}^{\mathrm{res}}$, we obtain

$$\varepsilon(K) = \mathrm{Tr}\left(W_{2,P}\Sigma_{P|K}^{\mathrm{res}}W_{2,P}^\top\right), \tag{C.39}$$

which proves Eq. equation 10.

Residual variance of one neuron. Let K be any set of kept neurons with $\Sigma_{KK} \succ 0$, and consider removing neuron $i \in K$. We ask how well its activation h_i can be predicted from the other surviving activations $h_{K\setminus\{i\}}$. Since the variance is invariant to the mean, we consider the linear predictor $\alpha^\top h_{K\setminus\{i\}}$ and solve

$$\min_\alpha \mathrm{Var}\left(h_i - \alpha^\top h_{K\setminus\{i\}}\right). \tag{C.40}$$

This is a particular case of the preceding affine reconstruction problem, with h_i as the scalar target and $h_{K\setminus\{i\}}$ as the predictors. Specialising Eq. equation C.28 gives

$$\alpha^\star = \Sigma_{K\setminus\{i\},K\setminus\{i\}}^{-1}\Sigma_{K\setminus\{i\},i}. \tag{C.41}$$

Substituting $\alpha^\star$ back into the objective yields

$$\min_\alpha \mathrm{Var}\left(h_i - \alpha^\top h_{K\setminus\{i\}}\right) = \Sigma_{ii} - \Sigma_{i,K\setminus\{i\}}\Sigma_{K\setminus\{i\},K\setminus\{i\}}^{-1}\Sigma_{K\setminus\{i\},i}. \tag{C.42}$$

This is the variance of the part of h_i that cannot be linearly reconstructed from the other surviving neurons.

To connect this quantity to the precision matrix, we write the covariance of the surviving activations by isolating neuron i:

$$\Sigma_{KK} = \begin{pmatrix} \Sigma_{ii} & \Sigma_{i,K\setminus\{i\}} \\ \Sigma_{K\setminus\{i\},i} & \Sigma_{K\setminus\{i\},K\setminus\{i\}} \end{pmatrix}. \tag{C.43}$$

By the block-inverse formula, the corresponding diagonal entry of $\Omega = \Sigma_{KK}^{-1}$ (where $\Sigma_{KK} = \mathrm{Cov}(h_K)$) is

$$\Omega_{ii} = \left(\Sigma_{ii} - \Sigma_{i,K\setminus\{i\}}\Sigma_{K\setminus\{i\},K\setminus\{i\}}^{-1}\Sigma_{K\setminus\{i\},i}\right)^{-1}. \tag{C.44}$$

Therefore,

$$\min_\alpha \mathrm{Var}\left(h_i - \alpha^\top h_{K\setminus\{i\}}\right) = \frac{1}{\Omega_{ii}}. \tag{C.45}$$

Hence, $1/\Omega_{ii}$ is exactly the residual variance of neuron i after its best linear reconstruction from the other surviving neurons. A small value means a small absolute reconstruction error, due to predictability from the remaining activations or a small marginal variance.

D EXACT UPDATES

Appendix C compensates a *fixed* set of kept neurons. This appendix proves Proposition 2 and Corollary 3, on which the greedy elimination of Section 3.1 rests. We again work within one block and omit the block index. We assume fixed hidden features with finite second moments and a positive-definite covariance Σ.

The cost of one removal. Suppose the elimination has reached the kept set K^t, the neurons already removed having been absorbed by the surviving neurons, so that the current compensated output is $\sum_{i\in K^t} w_i^t h_i + \widetilde{c}_2^t$, with the w_i^t the columns of the compensated second layer of Proposition 1 on K^t.

By equation C.29 and equation C.30, the reconstruction error already committed is orthogonal in L^2 to every affine function of h_{K^t}, hence to the change caused by removing one further neuron; the two squared errors therefore add. Removing $i \in K^t$ and re-optimising is an instance of Appendix C with kept set $K^t \setminus \{i\}$ and pruned set $\{i\}$, for which the residual covariance Σ^{res} is the scalar equation C.45. By equation C.39, the error introduced as a result is

$$\begin{aligned}\operatorname{Tr}\left(w_i^t\,\Sigma^{\mathrm{res}}\,{w_i^t}^\top\right) &= \|w_i^t\|_2^2\ \min_\alpha \operatorname{Var}\left(h_i - \alpha^\top h_{K^t\setminus\{i\}}\right)\\ &= \frac{\|w_i^t\|_2^2}{\Omega_{ii}^t},\end{aligned} \tag{D.46}$$

which proves Proposition 2. Both factors are already available from the quantities maintained by the elimination procedure.

The rank-one updates. Let j_t be the neuron removed at step t and $K^{t+1} = K^t \setminus \{j_t\}$. We partition $\Sigma_{K^tK^t}$ along (K^{t+1}, j_t) and abbreviate its three blocks as

$$S = \Sigma_{K^{t+1}K^{t+1}}, \qquad u = \Sigma_{K^{t+1}j_t}, \qquad v = \Sigma_{j_tj_t}. \tag{D.47}$$

The block-inversion formula for a symmetric matrix then gives

$$\Sigma_{K^tK^t} = \begin{pmatrix} S & u \\ u^\top & v \end{pmatrix}, \qquad \Omega^t = \begin{pmatrix} S^{-1} + \dfrac{S^{-1}uu^\top S^{-1}}{\sigma_{j_t}^2} & -\dfrac{S^{-1}u}{\sigma_{j_t}^2} \\ -\dfrac{u^\top S^{-1}}{\sigma_{j_t}^2} & \dfrac{1}{\sigma_{j_t}^2} \end{pmatrix}, \tag{D.48}$$

where $\sigma_{j_t}^2 = v - u^\top S^{-1}u$ is the Schur complement at j_t. Three readings of equation D.48 are all that the elimination needs.

The bottom-right entry gives $\Omega_{j_tj_t}^t = 1/\sigma_{j_t}^2$, which is equation C.45 specialised to the current kept set.

The off-diagonal block gives $\Omega_{K^{t+1}j_t}^t = -S^{-1}u\,\Omega_{j_tj_t}^t$, that is

$$\beta^{(j_t)} = \Sigma_{K^{t+1}K^{t+1}}^{-1}\Sigma_{K^{t+1}j_t} = -\frac{\Omega_{K^{t+1}j_t}^t}{\Omega_{j_tj_t}^t}, \tag{D.49}$$

which is the coefficient $-\Omega_{ij_t}^t/\Omega_{j_tj_t}^t$ appearing in equation 16: the regression of h_{j_t} on the surviving activations is already present in Ω^t, and costs a column read rather than a solve. Absorbing it by equation 9, with $K = K^{t+1}$, $P = \{j_t\}$ and $W_{2,P} = w_{j_t}^t$, gives the weight update equation 16 and the corresponding bias update from equation 9.

The upper-left block gives $\Omega_{K^{t+1}K^{t+1}}^t = S^{-1} + \Omega_{K^{t+1}j_t}^t\Omega_{j_tK^{t+1}}^t/\Omega_{j_tj_t}^t$. Rearranging,

$$\Omega^{t+1} = \Sigma_{K^{t+1}K^{t+1}}^{-1} = \Omega_{K^{t+1}K^{t+1}}^t - \frac{\Omega_{K^{t+1}j_t}^t\Omega_{j_tK^{t+1}}^t}{\Omega_{j_tj_t}^t}, \tag{D.50}$$

which is equation 17: the precision matrix of the new kept set is a rank-one downdate of the old one. No covariance is re-estimated and no matrix is re-inverted along the elimination.

The sequential compensations telescope.

Corollary 3. *Let $K^0 \supset K^1 \supset \cdots \supset K^m = K$ be an elimination chain, starting from $K^0 = \{1, \dots, H\}$, each set obtained from the previous one by removing a single neuron j_t. Then composing the one-step compensations of Proposition 1 along the chain yields the joint compensation on K, and the costs charged along the way sum to the total error:*

$$\varepsilon(K) = \sum_{t=0}^{m-1} \frac{\|w_{j_t}^t\|_2^2}{\Omega_{j_t j_t}^t}. \tag{D.51}$$

Proof of Corollary 3. For an index set J, let

$$\mathcal{A}_J = \left\{ a^\top h_J + a_0 \ : \ a \in \mathbb{R}^{|J|},\ a_0 \in \mathbb{R} \right\} \tag{D.52}$$

denote the linear span of the constant 1 and the activations indexed by J, a closed subspace of L^2. Since $\|y - \widetilde{y}\|_2^2 = \sum_{d=1}^D (y_d - \widetilde{y}_d)^2$ decouples across output coordinates, and $\widetilde{y}_d$ ranges over exactly $\mathcal{A}_K$ as $(\widetilde{W}_2, \widetilde{c}_2)$ vary, the optimal compensation of Appendix C is a coordinatewise orthogonal projection in L^2:

$$\widetilde{y}_d^\star = \Pi_{\mathcal{A}_K} y_d, \qquad \varepsilon(K) = \sum_{d=1}^D \|y_d - \Pi_{\mathcal{A}_K} y_d\|_{L^2}^2 . \tag{D.53}$$

For $J \subseteq J'$ we have $\mathcal{A}_J \subseteq \mathcal{A}_{J'}$, hence $\Pi_{\mathcal{A}_J} \Pi_{\mathcal{A}_{J'}} = \Pi_{\mathcal{A}_J}$. Composing the one-step compensations along the chain therefore yields $\Pi_{\mathcal{A}_{K^m}} \cdots \Pi_{\mathcal{A}_{K^1}} \Pi_{\mathcal{A}_{K^0}} = \Pi_{\mathcal{A}_K}$: the sequential compensations reproduce the joint solution of equation 8 and equation 9 exactly, which is why the running weights w_i^t are the columns of the jointly compensated $\widetilde{W}_2$ and not an approximation of them.

The same nesting makes the accounting exact. By Pythagoras applied to the nested projections,

$$\left\|y_d - \Pi_{\mathcal{A}_{K^{t+1}}} y_d\right\|_{L^2}^2 - \left\|y_d - \Pi_{\mathcal{A}_{K^t}} y_d\right\|_{L^2}^2 = \left\|\Pi_{\mathcal{A}_{K^t}} y_d - \Pi_{\mathcal{A}_{K^{t+1}}} y_d\right\|_{L^2}^2 , \tag{D.54}$$

whose right-hand side, summed over output coordinates d, is exactly the quantity equation D.46 evaluated on the state left by the previous compensations. Since $\varepsilon(K^0) = 0$, summing over the chain telescopes and gives equation D.51.

Remark: additivity along the elimination chain. The successive errors $\delta_t = \Pi_{\mathcal{A}_{K^t}} y - \Pi_{\mathcal{A}_{K^{t+1}}} y$ are mutually orthogonal in L^2: δ_t is a projection residual, hence orthogonal to $\mathcal{A}_{K^{t+1}}$, while every later δ_s, $s > t$, lies inside $\mathcal{A}_{K^s} \subseteq \mathcal{A}_{K^{t+1}}$. The decomposition $y - \Pi_{\mathcal{A}_K} y = \sum_t \delta_t$ is therefore orthogonal, so the squared errors add without cross terms, as in equation D.51.

This additivity holds for costs evaluated along the updated elimination chain; costs evaluated at a fixed state are not necessarily additive. Writing $s_i(K) = \|w_i^{(K)}\|_2^2 / \Omega_{ii}^{(K)}$ for the cost of i at kept set K, two removals give

$$\varepsilon\left(K \setminus \{i_1, i_2\}\right) - \varepsilon(K) = s_{i_1}(K) + s_{i_2}\left(K \setminus \{i_1\}\right) \overset{\text{in general}}{\neq} s_{i_1}(K) + s_{i_2}(K), \tag{D.55}$$

so the interaction between removals is accounted for by updating the second cost. Exact error accounting does not imply that greedy elimination finds the optimal kept set: the selection problem remains NP-hard (Appendix F).

E ARCHITECTURAL VARIANTS

This appendix shows how CARVE handles two common MLP variants: a learned gain applied after the MLP, such as LayerScale, and SwiGLU MLPs.

Post-MLP scaling (LayerScale). A fixed per-channel scale $g \in \mathbb{R}^D$, such as LayerScale in DeiT-III or the corresponding parameter in ConvNeXt, changes the residual contribution to

$$z^b = g \odot y^b = g \odot (W_2^b h^b + c_2^b). \tag{E.56}$$

Omitting the block index, the local reconstruction objective becomes

$$\min_{\widetilde{W}_2,\widetilde{c}_2} \mathbb{E}\|z-\widetilde{z}\|_2^2 = \min_{\widetilde{W}_2,\widetilde{c}_2} \mathbb{E}\|g\odot(y-\widetilde{y})\|_2^2. \tag{E.57}$$

With $\widehat{W}_2 = \mathrm{diag}(g)W_2$ and $\widehat{c}_2 = \mathrm{diag}(g)c_2$, Appendix C gives

$$\begin{aligned} \mathrm{diag}(g)\widetilde{W}_2^\star &= \mathrm{diag}(g)(W_{2,K} + W_{2,P}\beta^\top), \\ \mathrm{diag}(g)\widetilde{c}_2^\star &= \mathrm{diag}(g)(c_2 + W_{2,P}\mu_P - W_{2,P}\beta^\top\mu_K), \end{aligned} \tag{E.58}$$

where $\beta = \Sigma_{KK}^{-1}\Sigma_{KP}$. Thus the original compensation equation 9 remains optimal, including when a scale entry vanishes; the corresponding output parameters are then not uniquely determined. The resulting reconstruction error is

$$\varepsilon(K) = \mathrm{Tr}\left(\widehat{W}_{2,P}\Sigma_{P|K}^{\mathrm{res}}\widehat{W}_{2,P}^\top\right). \tag{E.59}$$

The one-removal cost and selection rule become

$$\begin{aligned} \varepsilon(K^t\setminus\{i\}) - \varepsilon(K^t) &= \frac{\|g\odot w_i^t\|_2^2}{\Omega_{ii}^t}, \\ j_t &= \arg\min_{i\in K^t} \frac{\|g\odot w_i^t\|_2^2}{\Omega_{ii}^t}. \end{aligned} \tag{E.60}$$

For $i \in K^{t+1} = K^t \setminus \{j_t\}$, the weight update satisfies

$$\begin{aligned} w_i^{t+1} &= w_i^t - \frac{\Omega_{ij_t}^t}{\Omega_{j_tj_t}^t} w_{j_t}^t, \\ g\odot w_i^{t+1} &= g\odot w_i^t - \frac{\Omega_{ij_t}^t}{\Omega_{j_tj_t}^t}(g\odot w_{j_t}^t). \end{aligned} \tag{E.61}$$

The original weights are therefore updated as before, with scores evaluated on their scaled columns. The precision downdate equation 17 is unchanged; for global allocation the modified removal cost is divided by V_b as in equation 20. Setting $g = \mathbf{1}$ recovers the original formulas.

SwiGLU MLPs. For SwiGLU, the hidden representation is

$$h = \mathrm{SiLU}(W_1x + c_1)\odot(Ux + c_u), \qquad y = W_2h + c_2, \tag{E.62}$$

where $W_1, U \in \mathbb{R}^{H\times D}$. Calibration measures the mean and covariance of this product h. Since Appendices C and D depend only on these statistics and W_2, their compensation, removal scores and updates apply unchanged.

Keeping indices K applies the same row selection to both input projections:

$$\begin{aligned} W_1 &\longrightarrow W_1[K,:], \quad & U &\longrightarrow U[K,:], \\ c_1 &\longrightarrow c_1[K], & c_u &\longrightarrow c_u[K], \\ W_2 &\longrightarrow \widetilde{W}_2, & c_2 &\longrightarrow \widetilde{c}_2. \end{aligned} \tag{E.63}$$

Biases are sliced when present, and $(\widetilde{W}_2, \widetilde{c}_2)$ are computed by equation 9. Each removed coordinate therefore removes three length-D weight vectors rather than two in a single-projection MLP. This is the variant used by Qwen2.5-1.5B in Appendix M.

F Column subset selection

This appendix expresses equation 6 as a column subset selection problem and establishes NP-hardness in general. The block index is omitted.

Output-weighted column subset selection. Let $A \in \mathbb{R}^{N\times H}$ contain the centred calibration activations, so that $\Sigma = A^\top A/N$. For expectations over these observations, Appendix C gives

$$\begin{aligned}\varepsilon_k^\star &= \min_{\substack{|K|=k\\ \widetilde{W}_2,\widetilde{c}_2}} \mathbb{E}\|y - \widetilde{W}_2 h_K - \widetilde{c}_2\|_2^2 \\ &= \min_{|K|=k} \varepsilon(K) = \min_{|K|=k} \operatorname{Tr}\Big(W_{2,P}\Sigma_{P|K}^{\mathrm{res}} W_{2,P}^\top\Big) \\ &= \min_{\substack{|K|=k\\ \widetilde{W}_2}} \frac{1}{N}\|AW_2^\top - A_K\widetilde{W}_2^\top\|_F^2.\end{aligned} \tag{F.64}$$

This is an instance of generalised column subset selection: columns of A are selected to reconstruct the target $AW_2^\top$. For singular covariances, the same least-squares formulation uses the Moore-Penrose pseudoinverse in $\Sigma_{P|K}^{\mathrm{res}}$.

Reduction from classical column subset selection. For a matrix $X \in \mathbb{R}^{n_0\times H}$, classical column subset selection minimises $\min_{|K|=k}\min_{\Theta\in\mathbb{R}^{k\times H}}\|X - X_K\Theta\|_F^2$ and is NP-hard (Shitov, 2021). Given such an instance, we construct

$$A = \begin{pmatrix} X \\ -X \end{pmatrix}, \qquad N = 2n_0, \qquad D = H, \qquad W_2 = I_H. \tag{F.65}$$

Every column of A has zero mean, and for every K and Θ, we have

$$\frac{1}{2n_0}\|A - A_K\Theta\|_F^2 = \frac{1}{n_0}\|X - X_K\Theta\|_F^2. \tag{F.66}$$

Taking $\widetilde{W}_2^\top = \Theta$, the construction preserves the optimal kept sets and is polynomial in the input size. An exact solver for the general output-weighted problem would therefore solve classical column subset selection. This establishes hardness for arbitrary output weights and dimensions, allowing singular covariances; it does not separately establish hardness for a fixed MLP expansion ratio or under the positive-definiteness assumption used by the inverse formulas.

Subset selection algorithms. Backward elimination for least-squares subset selection was studied by Couvreur & Bresler (2000), who established recovery guarantees under full-column-rank and sufficiently small perturbation assumptions. For forward greedy selection, Das & Kempe (2011) derived approximation guarantees using the submodularity ratio and sparse eigenvalues; Elenberg et al. (2018) related restricted strong convexity and smoothness to weak submodularity and greedy approximation guarantees. For classical matrix reconstruction, Deshpande et al. (2006) developed volume sampling and adaptive sampling with low-rank approximation guarantees. These results concern different algorithms or require additional assumptions; they do not directly establish a guarantee for the output-weighted backward elimination used here. CARVE uses the exact one-removal costs of Appendix D, while the choice of the final kept set remains heuristic.

II Additional experiments

G Numerical validation

In this appendix, we empirically validate the theoretical results of Section 3.1. Every quantity the method computes in closed form can also be measured, during or after pruning, with one additional forward pass over the calibration set. The two may differ slightly for two reasons: floating-point error, which can accumulate and be amplified along the elimination, and the ridge that regularises the covariance. Throughout, $\Sigma_\lambda = \Sigma + \lambda I$ denotes the regularised covariance of equation 21. The quantities $\Sigma_{P|K}^{\mathrm{res},\lambda}$, β_λ, and $(\widetilde{W}_2, \widetilde{c}_2)$ are, respectively, $\Sigma_{P|K}^{\mathrm{res}}$ of equation 10, β of equation 8, and the compensation of equation 9, all computed from Σ_λ.

For a set of tokens $\mathcal{D}$, the measured error of the pruned block is

$$\widehat{\varepsilon}_{\mathcal{D}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \big\| \widetilde{y}(x) - y(x) \big\|_2^2,$$

obtained by a forward pass of the teacher block of equation 1 and of the pruned block of equation 5 on the same teacher input x, where $\mathcal{C}$ denotes the calibration tokens and $\mathcal{U}$ the unseen tokens, those of $5{,}000$ other training images, disjoint from the calibration images.

We measure the gap for the following quantities, one per row of Table G.1:

- *The output mean is preserved.* Proposition 1 guarantees $\mathbb{E}[\widetilde{y}] = \mathbb{E}[y]$. From equation 1 and equation 5, we compute from the returned weights and the calibration mean μ

  $$\big\| \mathbb{E}[\widetilde{y}] - \mathbb{E}[y] \big\|_2 = \big\| \widetilde{W}_2 \mu_K + \widetilde{c}_2 - W_2 \mu - c_2 \big\|_2,$$

  which should be zero.
- *The scores add up.* Each score $\|w_{j_t}^t\|_2^2 / \Omega_{j_t j_t}^t$ of equation 15 is the error increment of one removal (Proposition 2), and successive increments are orthogonal, so the scores of the removed neurons should sum to the error of the final kept set given by equation 10. We therefore compare the sum of the scores of the removed neurons, where Ω^t is obtained from Ω^0 of equation 21 by the successive downdates of equation 17, with the error of the final kept set computed directly in a single solve:

  $$\sum_{t=0}^{|P|-1} \frac{\|w_{j_t}^t\|_2^2}{\Omega_{j_t j_t}^t} \quad \text{vs.} \quad \varepsilon^{\lambda}(K) = \mathrm{Tr}\big(W_{2,P}\, \Sigma_{P|K}^{\mathrm{res},\lambda}\, W_{2,P}^{\top}\big).$$

  Here $\varepsilon^{\lambda}(K)$ is the error equation 10 computed from Σ_λ, while $\varepsilon(K)$ keeps its ridge-free definition. Both sides are computed from Σ_λ, so this row checks the algebra of the elimination only.
- *Impact of the ridge.* The error the method predicts is the sum of the scores, i.e. by the previous item $\varepsilon^{\lambda}(K)$. We compare it with the drift measured at the output of the pruned block:

  $$\varepsilon^{\lambda}(K) = \mathrm{Tr}\big(W_{2,P}\, \Sigma_{P|K}^{\mathrm{res},\lambda}\, W_{2,P}^{\top}\big) \quad \text{vs.} \quad \widehat{\varepsilon}_{\mathcal{C}}.$$

  This gap is not expected to vanish: the prediction is computed under Σ_λ, whereas the calibration tokens have covariance Σ, so the prediction overestimates the error by a positive term of order ρ.
- *Impact of the ridge, corrected.* Using $(\Sigma_\lambda)_{KK} \beta_\lambda = \Sigma_{KP}$, the exact error of the returned block on the calibration tokens is $\mathrm{Tr}\big(W_{2,P}(\Sigma_{P|K}^{\mathrm{res},\lambda} - \lambda(I + \beta_\lambda^{\top}\beta_\lambda)) W_{2,P}^{\top}\big)$, so the prediction exceeds it by exactly the penalty

  $$\pi_\lambda = \lambda\big(\|W_{2,P}\|_F^2 + \|W_{2,P}\beta_\lambda^{\top}\|_F^2\big).$$

  We subtract this penalty from the prediction and compare again with the measured drift:

  $$\varepsilon^{\lambda}(K) - \pi_\lambda \quad \text{vs.} \quad \widehat{\varepsilon}_{\mathcal{C}}.$$

  This gap should be much smaller and explained only by the numerical errors from the `float32` forward pass.
- *Error increase due to the ridge.* The theory states the error at $\lambda = 0$, with the optimal β of equation 8, whereas the returned block uses β_λ. We compare the best error achievable on the same kept set with the error measured for the regularised block:

  $$\varepsilon(K) = \mathrm{Tr}\big(W_{2,P}\, \Sigma_{P|K}^{\mathrm{res}}\, W_{2,P}^{\top}\big) \quad \text{vs.} \quad \widehat{\varepsilon}_{\mathcal{C}}.$$

  Since β minimises the error, the first is always the smaller, and the gap is the increase in mean squared error due to the regularisation.
- *Calibration shift.* We repeat the measurement on the unseen tokens $\mathcal{U}$, which the statistics never saw:

  $$\widehat{\varepsilon}_{\mathcal{U}} \quad \text{vs.} \quad \widehat{\varepsilon}_{\mathcal{C}}.$$

  This gap is the sampling error of μ and Σ.

Table G.1 shows the results obtained using DeiT-Base on the ImageNet-1k dataset, and all quantities are summarised by their median over all 12 blocks and 9 sparsities. On DeiT-Base, the calibration covariance is never singular: the smallest eigenvalue in the experiment was 8×10^{-6} times its mean diagonal, so the ridge-free quantity of the fifth row is well defined. This is not the case on every model: on ViT-B/16, ConvNeXt-Base, DeiT-III-Large and DeiT-III-Huge, the covariance of some blocks is singular, and the ridge is required. The unseen tokens $\mathcal{U}$ come from a class-balanced draw of $5{,}000$ training images, disjoint from the calibration images.

Table G.1: Error measurement on DeiT-Base on ImageNet-1k, median over the 108 (block, width) pairs.

Experiment	Quantity measured	Relative median error
Mean preservation	$\lVert\widetilde{W}_2\mu_K + \widetilde{c}_2 - W_2\mu - c_2\rVert_2$	1.0×10^{-7}
Algebra validation	$\sum_t \lVert w_{j_t}^t\rVert_2^2/\Omega_{j_t j_t}^t - \varepsilon^\lambda(K)$	1.0×10^{-15}
Impact of the ridge	$\varepsilon^\lambda(K) - \widehat{\varepsilon}_{\mathcal{C}}$	5.8×10^{-3}
Impact of the ridge, corrected	$\varepsilon^\lambda(K) - \pi_\lambda - \widehat{\varepsilon}_{\mathcal{C}}$	5.0×10^{-9}
Error increase due to the ridge	$\widehat{\varepsilon}_{\mathcal{C}} - \varepsilon(K)$	4.6×10^{-6}
Calibration shift	$\widehat{\varepsilon}_{\mathcal{U}} - \widehat{\varepsilon}_{\mathcal{C}}$	6.1×10^{-3}

The first two rows empirically validate the formula despite score accumulation over up to $2{,}765$ successive downdates. The ridge causes a 0.6% overestimation of the error; subtracting the expected penalty π_λ reduces the relative discrepancy to 5×10^{-9}, consistent with `float32` forward-pass noise, indicating that the ridge accounts for the entire gap. Appendix H further shows that this bias decreases proportionally to ρ. In contrast, the ridge has a negligible effect on the pruned block itself, whose error is only 5×10^{-6} above the ridge-free optimum. Finally, the median gap between unseen and calibration token errors is 0.6%, comparable to the ridge-induced prediction bias.

H RIDGE SENSITIVITY

This appendix measures how the ridge ρ affects accuracy and the error accounting of Appendix G. A single constant, $\rho = 10^{-3}$, regularises every solve. For each block, the ridge $\lambda = \rho\,\overline{\mathrm{diag}(\Sigma)}$ is set once from the full covariance: the elimination inverts Σ_λ once and the compensation of equation 9 uses the same λ on the kept block.

Table H.1: ImageNet-1k Top-1 (%) on DeiT-Base for different values of ρ.

ρ	MLP sparsity (%)								
	10	20	30	40	50	60	70	80	90
0	81.7	81.6	81.0	79.7	77.4	74.3	69.3	60.9	44.3
10^{-6}	81.7	81.6	81.0	79.6	77.5	74.3	69.3	60.9	44.3
10^{-5}	81.7	81.6	81.0	79.7	77.4	74.3	69.3	60.8	44.3
10^{-4}	81.7	81.6	80.9	79.6	77.4	74.4	69.3	60.9	44.2
10^{-3}	81.7	81.6	81.1	79.6	77.3	74.1	69.1	60.7	44.1
10^{-2}	81.7	81.6	81.1	79.8	77.5	74.2	69.3	61.0	43.9
10^{-1}	81.6	81.4	81.3	80.5	78.2	74.9	69.7	60.7	43.6

From $\rho = 0$ (no regularisation) to 10^{-3}, the accuracy remains almost constant, and $\rho = 10^{-2}$ changes no accuracy by more than 0.4 points. Only $\rho = 10^{-1}$ changes the results, and a stronger regularisation improves or worsens the results depending on the sparsity. The value $\rho = 10^{-3}$ is not tuned, but its impact is small.

Table H.2: Relative impact of the ridge on the error estimation, $\pi_\lambda/\widehat{\varepsilon}_{\mathcal{C}}$ (third row of Table G.1).

ρ	block 0	block 5	block 11	worst of 12
10^{-3}	3.3	1.4×10^{-1}	2.3×10^{-2}	3.3
10^{-4}	3.7×10^{-1}	1.4×10^{-2}	2.3×10^{-3}	3.7×10^{-1}
10^{-5}	3.8×10^{-2}	1.5×10^{-3}	2.3×10^{-4}	3.8×10^{-2}
10^{-6}	3.8×10^{-3}	1.5×10^{-4}	2.3×10^{-5}	3.8×10^{-3}

Table H.2 shows that the ridge penalty falls by a factor of ten per decade of ρ, in every block. The gap between predicted and measured error is therefore a single term linear in ρ, and the identities of Section 3.1 become exact as $\rho \to 0$.

I Ablation: normalisation

In this appendix we study the impact of the normalisation on the global allocation and the resulting performance. We hold each block's neuron ordering and affine compensation fixed and vary only how the same total neuron budget is distributed across blocks.

Global allocation rules. VBP and DVBP rank MLP neurons by post-activation variance and use those scores across all blocks, letting a single global threshold set each block's width (Berisha et al., 2025; Lee et al., 2026). CARVE_{G} applies the same rule to CARVE's score: it uses local scores globally without normalisation. CARVE_{L} and VBP_{L} remove the allocation entirely by fixing a constant per-block width.

Table I.1: ImageNet-1k, 80% MLP sparsity. V_b is the variance in the input residual stream.

Budget rule	DeiT-T	DeiT-S	DeiT-B	ViT-B/16	DeiT-III-L	DeiT-III-H
Raw costs, unnormalised (CARVE_{G})	20.9	44.3	34.8	31.2	25.4	46.1
Uniform per-block sparsity (CARVE_{L})	15.5	44.3	57.6	51.4	71.4	77.8
Divide by stream variance V_b (CARVE)	**22.6**	**52.0**	**60.7**	**53.2**	**72.2**	**78.1**

Table I.1 compares the three allocation rules on the six Transformer backbones at 80% MLP sparsity. At 80% sparsity, the unnormalised allocator is worse than uniform allocation on five of the six Transformers, marginally so on DeiT-Small and by 22.8, 46.0 and 31.7 Top-1 points on DeiT-Base, DeiT-III-Large and DeiT-III-Huge. Since selection, reconstruction and total width are unchanged, the allocation alone causes the collapse. The failure is also model-dependent: on DeiT-Tiny the unnormalised allocator is the better of the two, so there is no single sparsity at which it stops working.

Impact of the normalisation. Figure I.1 compares, on DeiT-Base, each score under a global allocation and at a constant per-block sparsity. We observe that CARVE exceeds CARVE_{L} at every sparsity: by 0.8 points at 10%, by 2.6 to 3.1 points from 30% to 80%, and by 4.6 points at 90%. This also demonstrates that using unnormalised scores across blocks helps at mild sparsity, by up to 5.8 points for VBP and 3.5 for CARVE at 40%. Both curves, however, fall below their uniform-sparsity control between 50% and 60% sparsity and then collapse: at 80%, VBP loses 14.2 points against VBP_{L}, and CARVE_{G} loses 22.8 against CARVE_{L}. Dividing by V_b removes this collapse while keeping most of the mild-sparsity gain.

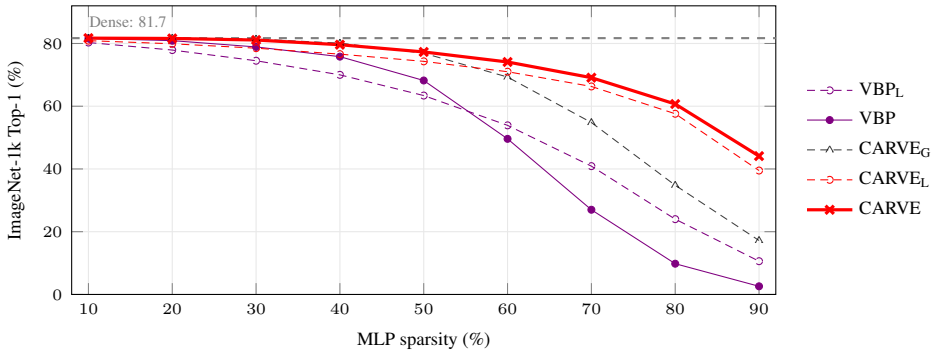

Figure I.1: ImageNet-1k Top-1 accuracy at increasing sparsity on DeiT-Base.

J IMAGENET-1K RESULTS

Table J.1 reports the complete grid behind Table 1. Below 20% sparsity, compensated methods are close to the teacher, but their performance gaps widen as sparsity increases. MAG collapses early, whereas SNIP degrades more gradually and remains competitive up to higher sparsity levels on deeper backbones.

Table J.1: Top-1 (%) on ImageNet-1k across the nine vision backbones and nine MLP sparsities, one-shot and without fine-tuning. Mean ± standard deviation over three calibration seeds; MAG is deterministic and is run once. Best result in **bold**, second best underlined. Teacher, competing pruning methods, and our method.

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
DeiT-Tiny								
	0	Teacher: 72.0						
	10	63.1	69.1 ±0.2	70.9 ±0.0	70.9 ±0.0	**71.5** ±0.0	**71.5** ±0.0	71.2 ±0.0
	20	40.8	61.2 ±0.4	68.9 ±0.1	69.2 ±0.0	69.7 ±0.0	**70.1** ±0.0	69.7 ±0.0
	30	15.5	46.6 ±1.8	65.1 ±0.1	66.5 ±0.1	66.5 ±0.0	67.1 ±0.0	**68.4** ±0.0
	40	3.3	27.1 ±1.5	57.0 ±0.1	62.0 ±0.0	62.0 ±0.1	63.0 ±0.0	**65.5** ±0.0
	50	0.5	12.0 ±1.0	42.2 ±0.3	55.9 ±0.1	55.3 ±0.1	56.6 ±0.1	**59.7** ±0.1
	60	0.3	4.4 ±0.4	27.5 ±0.1	47.2 ±0.1	45.2 ±0.2	46.7 ±0.3	**51.2** ±0.1
	70	0.2	1.8 ±0.2	15.3 ±0.3	34.4 ±0.3	28.9 ±0.2	31.0 ±0.1	**38.2** ±0.1
	80	0.1	0.7 ±0.2	6.2 ±0.0	16.7 ±0.5	10.6 ±0.4	12.0 ±0.3	**22.6** ±0.6
	90	0.2	0.3 ±0.1	2.4 ±0.0	4.4 ±0.1	2.2 ±0.0	1.7 ±0.1	**8.5** ±0.3
DeiT-Small								
	0	Teacher: 79.7						
	10	72.0	76.4 ±0.2	78.3 ±0.0	78.4 ±0.0	**78.5** ±0.0	78.2 ±0.1	78.3 ±0.0
	20	44.3	71.5 ±0.2	77.0 ±0.0	**77.6** ±0.0	77.2 ±0.0	77.2 ±0.0	**77.6** ±0.0
	30	12.8	63.0 ±0.8	75.1 ±0.0	75.8 ±0.0	75.5 ±0.1	75.7 ±0.1	**77.0** ±0.0
	40	1.7	53.0 ±1.1	72.0 ±0.1	73.4 ±0.1	72.8 ±0.1	73.4 ±0.0	**75.7** ±0.0
	50	0.4	39.9 ±1.4	65.7 ±0.2	69.7 ±0.1	69.2 ±0.1	70.2 ±0.1	**72.9** ±0.1
	60	0.2	23.5 ±2.7	53.3 ±0.2	64.5 ±0.1	63.5 ±0.1	64.9 ±0.1	**68.7** ±0.1
	70	0.1	10.8 ±0.6	34.0 ±0.4	56.3 ±0.2	53.9 ±0.1	55.3 ±0.1	**62.0** ±0.1
	80	0.1	2.5 ±0.4	15.6 ±0.4	38.5 ±0.2	36.0 ±0.1	37.7 ±0.2	**52.0** ±0.1
	90	0.2	0.7 ±0.1	6.7 ±0.1	15.3 ±0.2	9.7 ±0.3	7.2 ±0.5	**33.1** ±0.4
DeiT-Base								
	0	Teacher: 81.7						
	10	74.0	78.4 ±0.2	81.6 ±0.0	**81.7** ±0.0	80.7 ±0.1	80.8 ±0.1	**81.7** ±0.0
	20	47.6	73.8 ±0.2	80.9 ±0.0	**81.6** ±0.0	79.4 ±0.0	79.7 ±0.1	**81.6** ±0.0
	30	4.8	68.2 ±0.6	78.9 ±0.0	80.8 ±0.1	77.7 ±0.1	78.3 ±0.1	**81.1** ±0.1
	40	0.3	59.9 ±0.6	75.8 ±0.1	78.8 ±0.1	75.4 ±0.1	76.2 ±0.1	**79.6** ±0.0
	50	0.2	47.1 ±1.2	68.2 ±0.1	75.2 ±0.0	72.3 ±0.1	73.2 ±0.1	**77.3** ±0.1
	60	0.1	29.4 ±0.7	49.6 ±0.2	68.7 ±0.2	68.0 ±0.1	69.4 ±0.1	**74.1** ±0.1
	70	0.1	11.6 ±0.6	27.0 ±0.6	56.0 ±0.2	61.8 ±0.1	63.5 ±0.1	**69.1** ±0.1
	80	0.1	2.7 ±0.3	9.8 ±0.0	34.7 ±0.2	50.6 ±0.1	53.5 ±0.1	**60.7** ±0.1
	90	0.1	0.7 ±0.1	2.6 ±0.1	12.7 ±0.5	27.9 ±0.2	30.7 ±0.2	**44.1** ±0.1

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
ViT-B/16								
	0	**Teacher: 81.1**						
	10	30.4	78.9 ±0.2	80.4 ±0.0	80.9 ±0.0	80.6 ±0.0	80.7 ±0.0	**81.0** ±0.0
	20	3.9	74.4 ±0.2	79.6 ±0.0	80.7 ±0.0	79.7 ±0.0	79.8 ±0.1	**80.9** ±0.0
	30	0.4	65.9 ±0.5	77.3 ±0.0	79.4 ±0.0	78.4 ±0.1	78.5 ±0.0	**80.2** ±0.0
	40	0.1	53.5 ±0.8	71.3 ±0.1	76.8 ±0.0	76.5 ±0.1	76.7 ±0.1	**78.6** ±0.0
	50	0.2	36.4 ±0.4	59.9 ±0.2	72.8 ±0.1	73.8 ±0.1	74.0 ±0.1	**76.4** ±0.1
	60	0.1	15.2 ±0.4	36.8 ±0.3	66.0 ±0.1	69.1 ±0.2	69.7 ±0.2	**72.5** ±0.1
	70	0.1	3.3 ±0.4	13.3 ±0.1	54.2 ±0.1	60.8 ±0.1	61.4 ±0.2	**65.7** ±0.0
	80	0.2	0.7 ±0.0	3.3 ±0.1	36.2 ±0.1	46.2 ±0.3	46.4 ±0.3	**53.2** ±0.1
	90	0.1	0.3 ±0.0	0.9 ±0.0	9.3 ±0.1	20.6 ±0.3	20.1 ±0.1	**29.2** ±0.3
DeiT-III-Large								
	0	**Teacher: 84.6**						
	10	83.8	83.7 ±0.0	84.5 ±0.0	84.5 ±0.0	**84.6** ±0.0	**84.6** ±0.0	84.5 ±0.0
	20	78.3	82.1 ±0.1	84.3 ±0.0	**84.4** ±0.0	**84.4** ±0.0	**84.4** ±0.0	**84.4** ±0.0
	30	45.6	79.8 ±0.3	82.8 ±0.0	83.4 ±0.0	83.9 ±0.0	84.0 ±0.0	**84.4** ±0.0
	40	6.0	76.2 ±0.3	80.3 ±0.0	81.6 ±0.1	83.2 ±0.0	83.3 ±0.0	**84.0** ±0.0
	50	0.7	70.9 ±0.5	76.8 ±0.1	78.9 ±0.0	82.0 ±0.0	82.2 ±0.0	**83.0** ±0.0
	60	0.3	63.2 ±1.1	72.2 ±0.2	75.0 ±0.1	79.9 ±0.1	80.1 ±0.0	**81.2** ±0.0
	70	0.2	51.5 ±1.5	66.7 ±0.2	69.8 ±0.1	75.6 ±0.1	75.8 ±0.1	**77.9** ±0.1
	80	0.1	37.0 ±1.4	60.0 ±0.4	61.8 ±0.1	66.8 ±0.1	66.5 ±0.2	**72.2** ±0.1
	90	0.1	10.1 ±1.8	48.0 ±0.2	53.4 ±0.1	46.7 ±0.4	42.0 ±0.3	**57.9** ±0.3
DeiT-III-Huge								
	0	**Teacher: 85.0**						
	10	84.2	84.9 ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0
	20	81.1	84.6 ±0.0	84.9 ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0
	30	59.6	83.5 ±0.1	84.6 ±0.0	84.8 ±0.0	**84.9** ±0.0	**84.9** ±0.0	84.8 ±0.0
	40	5.2	81.5 ±0.1	83.5 ±0.0	84.1 ±0.0	84.7 ±0.0	84.7 ±0.0	**84.8** ±0.0
	50	0.6	78.7 ±0.2	81.1 ±0.0	82.5 ±0.1	84.1 ±0.0	84.2 ±0.1	**84.5** ±0.0
	60	0.1	73.6 ±0.2	78.5 ±0.1	80.4 ±0.1	83.1 ±0.0	83.1 ±0.1	**83.7** ±0.0
	70	0.1	64.8 ±0.8	74.3 ±0.0	76.6 ±0.0	80.8 ±0.0	80.9 ±0.0	**81.8** ±0.1
	80	0.1	52.9 ±1.1	67.8 ±0.1	71.0 ±0.1	75.4 ±0.0	75.6 ±0.0	**78.1** ±0.1
	90	0.2	25.4 ±0.9	58.0 ±0.3	61.4 ±0.1	59.6 ±0.1	57.0 ±0.3	**68.4** ±0.0

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
ConvNeXt-T								
	0	**Teacher: 84.0**						
	10	38.0	76.8 ±1.0	82.8 ±0.1	83.0 ±0.1	83.3 ±0.0	**83.5** ±0.0	83.0 ±0.0
	20	4.6	63.6 ±0.9	77.9 ±0.2	79.2 ±0.2	**81.8** ±0.0	**81.8** ±0.1	81.3 ±0.0
	30	0.4	35.3 ±2.8	63.3 ±0.2	69.1 ±0.2	78.6 ±0.0	**78.9** ±0.1	78.8 ±0.0
	40	0.1	6.6 ±0.4	36.0 ±0.9	53.6 ±0.2	72.5 ±0.2	73.1 ±0.0	**74.7** ±0.1
	50	0.1	1.0 ±0.1	14.0 ±0.3	35.0 ±0.8	60.5 ±0.0	62.3 ±0.2	**67.6** ±0.1
	60	0.1	0.3 ±0.0	2.8 ±0.1	17.0 ±0.6	41.4 ±0.3	42.1 ±0.5	**51.5** ±0.4
	70	0.1	0.3 ±0.0	0.5 ±0.1	10.0 ±0.1	19.6 ±0.5	17.5 ±0.8	**25.5** ±0.6
	80	0.1	0.1 ±0.0	1.6 ±0.0	3.5 ±0.3	4.0 ±0.1	3.2 ±0.1	**7.9** ±0.3
	90	0.1	0.1 ±0.0	0.5 ±0.0	0.6 ±0.1	0.4 ±0.0	0.2 ±0.0	**1.0** ±0.1
ConvNeXt-S								
	0	**Teacher: 84.9**						
	10	28.0	80.0 ±0.0	83.9 ±0.1	83.9 ±0.0	**84.5** ±0.0	**84.5** ±0.0	84.2 ±0.0
	20	3.1	63.8 ±1.2	81.2 ±0.1	81.6 ±0.0	83.1 ±0.0	**83.3** ±0.0	83.0 ±0.0
	30	0.4	25.8 ±0.8	73.5 ±0.2	75.3 ±0.2	80.0 ±0.0	80.6 ±0.0	**81.4** ±0.0
	40	0.1	4.0 ±0.5	52.8 ±0.5	60.2 ±0.4	74.4 ±0.1	75.4 ±0.1	**78.6** ±0.1
	50	0.1	0.8 ±0.2	27.0 ±1.0	40.7 ±0.6	63.3 ±0.1	64.7 ±0.2	**73.7** ±0.0
	60	0.1	0.3 ±0.1	9.9 ±0.7	21.7 ±0.6	45.5 ±0.2	46.4 ±0.2	**64.2** ±0.2
	70	0.1	0.2 ±0.1	3.5 ±0.3	9.4 ±0.8	24.5 ±0.3	24.7 ±0.2	**45.4** ±0.1
	80	0.1	0.1 ±0.0	2.7 ±0.1	3.7 ±0.4	5.6 ±0.2	5.1 ±0.1	**18.8** ±0.1
	90	0.1	0.1 ±0.0	0.4 ±0.0	1.0 ±0.2	0.6 ±0.0	0.5 ±0.0	**2.0** ±0.2
ConvNeXt-B								
	0	**Teacher: 85.5**						
	10	56.2	83.7 ±0.2	85.2 ±0.0	85.2 ±0.0	85.3 ±0.0	85.3 ±0.0	**85.4** ±0.0
	20	1.0	79.5 ±0.1	84.0 ±0.0	84.3 ±0.0	**84.7** ±0.0	**84.7** ±0.0	**84.7** ±0.1
	30	0.2	70.6 ±0.2	80.2 ±0.0	81.2 ±0.0	83.3 ±0.0	**83.5** ±0.0	**83.5** ±0.0
	40	0.1	36.0 ±0.8	71.5 ±0.1	74.8 ±0.1	80.7 ±0.1	80.9 ±0.1	**81.2** ±0.0
	50	0.1	2.9 ±0.7	56.5 ±0.4	64.5 ±0.5	75.4 ±0.1	75.8 ±0.1	**76.9** ±0.1
	60	0.1	0.6 ±0.2	39.9 ±0.2	50.8 ±0.4	63.7 ±0.1	64.3 ±0.1	**68.8** ±0.2
	70	0.1	0.3 ±0.1	24.6 ±0.2	33.3 ±0.3	43.4 ±0.4	41.5 ±0.6	**51.8** ±0.2
	80	0.1	0.1 ±0.0	10.2 ±0.5	19.2 ±1.2	18.0 ±0.2	14.6 ±0.4	**25.6** ±0.3
	90	0.1	0.1 ±0.0	1.9 ±0.1	**6.0** ±0.2	1.5 ±0.1	0.5 ±0.0	2.8 ±0.0

K GAP CLOSURE

This appendix details, for each backbone and each baseline, the gap recovery reported in Table 2 of Section 4.2.

For a backbone with teacher accuracy a_T and a baseline b with accuracy $a_b(s)$ at sparsity s, we measure the fraction of the remaining teacher gap recovered by CARVE as

$$\mathrm{Recovery}_b = \frac{\sum_s \big(a_{\mathrm{CARVE}}(s) - a_b(s)\big)}{\sum_s \big(a_T - a_b(s)\big)}.$$

The sums are taken over the nine MLP sparsities s from 10% to 90%, and each accuracy corresponds to the Top-1 accuracy reported in Table J.1, averaged over the three calibration seeds. For the *Best of the six* reference, $a_b(s)$ is replaced at each sparsity by the highest accuracy among the six competing baselines.

Table K.1: Share of each baseline's teacher gap recovered by CARVE (%), computed over the nine MLP sparsities of each backbone.

Backbone	MAG	SNIP	VBP	DVBP	CORP	GRAIL	Best of the six
DeiT-Tiny	63.1	54.5	33.9	12.5	18.2	15.4	11.0
DeiT-Small	79.5	68.0	49.9	28.4	33.6	32.2	28.0
DeiT-Base	85.8	76.3	67.0	47.9	39.2	33.8	28.3
ViT-B/16	83.9	72.1	63.5	35.4	22.3	21.3	19.9
DeiT-III-Large	90.5	74.9	51.0	41.4	30.1	34.0	22.4
DeiT-III-Huge	94.6	78.6	57.1	46.7	31.8	35.0	27.8
ConvNeXt-T	60.1	50.3	40.3	29.8	9.3	9.2	8.2
ConvNeXt-S	68.2	60.5	45.8	39.8	23.1	22.2	21.9
ConvNeXt-B	70.6	57.8	33.8	22.7	10.5	12.4	7.7

Table K.1 shows that CARVE recovers a positive share of the gap on every backbone and against every baseline. Against the best of the six baselines, it recovers between 11.0% (DeiT-Tiny) and 28.3% (DeiT-Base) of the remaining gap on the Transformers, and between 7.7% (ConvNeXt-B) and 21.9% (ConvNeXt-S) on ConvNeXt, where CORP and GRAIL are the strongest baselines.

L UNIFORM PER-BLOCK SPARSITY RESULTS

We apply every method with the same per-block sparsity, so only neuron selection and compensation vary. Table L.1 reports the results for every backbone and sparsity. $\mathrm{CARVE}_{\mathrm{L}}$ has the highest mean accuracy on every backbone.

Table L.1: Top-1 (%) on ImageNet-1k at uniform per-block sparsity, across the nine vision backbones and nine MLP sparsities. Only neuron selection and compensation vary. Mean ± standard deviation over three calibration seeds; MAG is deterministic and is run once. Best result in **bold**, second best underlined. Teacher , competing pruning methods , and our method .

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
DeiT-Tiny								
	0	**Teacher: 72.0**						
	10	63.1	69.1 ±0.2	70.8 ±0.0	71.3 ±0.0	**71.5** ±0.0	**71.5** ±0.0	**71.5** ±0.0
	20	40.8	61.2 ±0.4	66.6 ±0.1	68.7 ±0.1	69.7 ±0.0	**70.1** ±0.0	**70.1** ±0.0
	30	15.5	46.6 ±1.8	59.6 ±0.1	64.9 ±0.0	66.5 ±0.0	67.1 ±0.0	**67.6** ±0.0
	40	3.3	27.1 ±1.5	47.3 ±0.1	59.7 ±0.1	62.0 ±0.1	63.0 ±0.0	**63.9** ±0.0
	50	0.5	12.0 ±1.0	30.5 ±0.5	51.9 ±0.1	55.3 ±0.1	56.6 ±0.1	**58.0** ±0.0
	60	0.3	4.4 ±0.4	13.3 ±0.3	41.8 ±0.1	45.2 ±0.2	46.7 ±0.3	**48.8** ±0.3
	70	0.2	1.8 ±0.2	3.9 ±0.2	26.8 ±0.4	28.9 ±0.2	31.0 ±0.1	**35.3** ±0.2
	80	0.1	0.7 ±0.2	1.5 ±0.0	11.5 ±0.2	10.6 ±0.4	12.0 ±0.3	**15.5** ±0.3
	90	0.2	0.3 ±0.1	0.6 ±0.0	1.9 ±0.0	2.2 ±0.0	1.7 ±0.1	**3.1** ±0.1
DeiT-Small								
	0	**Teacher: 79.7**						
	10	72.0	76.4 ±0.2	78.0 ±0.0	78.3 ±0.0	78.5 ±0.0	78.2 ±0.1	**78.7** ±0.0
	20	44.3	71.5 ±0.2	75.4 ±0.0	76.4 ±0.0	77.2 ±0.0	77.2 ±0.0	**77.6** ±0.0
	30	12.8	63.0 ±0.8	71.7 ±0.1	74.2 ±0.0	75.5 ±0.1	75.7 ±0.1	**76.2** ±0.1
	40	1.7	53.0 ±1.1	65.4 ±0.1	71.1 ±0.0	72.8 ±0.1	73.4 ±0.0	**74.0** ±0.1
	50	0.4	39.9 ±1.4	57.2 ±0.2	67.1 ±0.1	69.2 ±0.1	70.2 ±0.1	**71.2** ±0.0
	60	0.2	23.5 ±2.7	43.7 ±0.1	61.1 ±0.1	63.5 ±0.1	64.9 ±0.1	**66.5** ±0.1
	70	0.1	10.8 ±0.6	22.9 ±0.1	51.2 ±0.0	53.9 ±0.1	55.3 ±0.1	**58.8** ±0.1
	80	0.1	2.5 ±0.4	6.6 ±0.3	31.6 ±0.4	36.0 ±0.1	37.7 ±0.2	**44.3** ±0.4
	90	0.2	0.7 ±0.1	1.1 ±0.0	8.2 ±0.1	9.7 ±0.3	7.2 ±0.5	**18.1** ±0.2
DeiT-Base								
	0	**Teacher: 81.7**						
	10	74.0	78.4 ±0.2	80.3 ±0.1	80.4 ±0.1	80.7 ±0.1	80.8 ±0.1	**80.9** ±0.0
	20	47.6	73.8 ±0.2	77.9 ±0.1	78.6 ±0.1	79.4 ±0.0	79.7 ±0.1	**79.9** ±0.1
	30	4.8	68.2 ±0.6	74.5 ±0.1	76.4 ±0.1	77.7 ±0.1	78.3 ±0.1	**78.5** ±0.1
	40	0.3	59.9 ±0.6	70.0 ±0.0	73.5 ±0.1	75.4 ±0.1	76.2 ±0.1	**76.6** ±0.1
	50	0.2	47.1 ±1.2	63.4 ±0.1	70.1 ±0.1	72.3 ±0.1	73.2 ±0.1	**74.3** ±0.1
	60	0.1	29.4 ±0.7	53.9 ±0.1	65.6 ±0.1	68.0 ±0.1	69.4 ±0.1	**71.0** ±0.1
	70	0.1	11.6 ±0.6	40.9 ±0.4	58.6 ±0.1	61.8 ±0.1	63.5 ±0.1	**66.3** ±0.2
	80	0.1	2.7 ±0.3	24.0 ±0.5	47.3 ±0.0	50.6 ±0.1	53.5 ±0.1	**57.6** ±0.1
	90	0.1	0.7 ±0.1	10.6 ±0.3	19.1 ±0.6	27.9 ±0.2	30.7 ±0.2	**39.5** ±0.2

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
ViT-B/16								
	0	**Teacher: 81.1**						
	10	30.4	78.9 ±0.2	80.3 ±0.0	80.5 ±0.0	80.6 ±0.0	**80.7** ±0.0	**80.7** ±0.0
	20	3.9	74.4 ±0.2	78.3 ±0.0	79.2 ±0.0	79.7 ±0.0	79.8 ±0.1	**80.0** ±0.0
	30	0.4	65.9 ±0.5	75.3 ±0.0	77.5 ±0.1	78.4 ±0.1	78.5 ±0.0	**78.9** ±0.1
	40	0.1	53.5 ±0.8	71.4 ±0.0	75.1 ±0.0	76.5 ±0.1	76.7 ±0.1	**77.2** ±0.1
	50	0.2	36.4 ±0.4	65.3 ±0.1	71.8 ±0.2	73.8 ±0.1	74.0 ±0.1	**74.7** ±0.1
	60	0.1	15.2 ±0.4	55.5 ±0.2	66.6 ±0.1	69.1 ±0.2	69.7 ±0.2	**71.2** ±0.1
	70	0.1	3.3 ±0.4	39.8 ±0.2	59.2 ±0.1	60.8 ±0.1	61.4 ±0.2	**64.3** ±0.1
	80	0.2	0.7 ±0.0	18.8 ±0.1	45.5 ±0.2	46.2 ±0.3	46.4 ±0.3	**51.4** ±0.3
	90	0.1	0.3 ±0.0	2.0 ±0.1	17.7 ±0.3	20.6 ±0.3	20.1 ±0.1	**28.5** ±0.5
DeiT-III-Large								
	0	**Teacher: 84.6**						
	10	83.8	83.7 ±0.0	84.2 ±0.0	84.5 ±0.0	**84.6** ±0.0	**84.6** ±0.0	**84.6** ±0.0
	20	78.3	82.1 ±0.1	83.6 ±0.0	84.2 ±0.0	**84.4** ±0.0	**84.4** ±0.0	**84.4** ±0.0
	30	45.6	79.8 ±0.3	82.6 ±0.0	83.5 ±0.0	83.9 ±0.0	84.0 ±0.0	**84.1** ±0.0
	40	6.0	76.2 ±0.3	81.2 ±0.0	82.5 ±0.1	83.2 ±0.0	83.3 ±0.0	**83.5** ±0.0
	50	0.7	70.9 ±0.5	79.0 ±0.1	81.0 ±0.1	82.0 ±0.0	82.2 ±0.0	**82.5** ±0.0
	60	0.3	63.2 ±1.1	73.8 ±0.1	78.4 ±0.1	79.9 ±0.1	80.1 ±0.0	**80.9** ±0.0
	70	0.2	51.5 ±1.5	62.3 ±0.2	73.1 ±0.0	75.6 ±0.1	75.8 ±0.1	**77.9** ±0.1
	80	0.1	37.0 ±1.4	43.6 ±0.1	63.8 ±0.1	66.8 ±0.1	66.5 ±0.2	**71.4** ±0.1
	90	0.1	10.1 ±1.8	22.1 ±0.7	45.7 ±0.1	46.7 ±0.4	42.0 ±0.3	**55.0** ±0.3
DeiT-III-Huge								
	0	**Teacher: 85.0**						
	10	84.2	84.9 ±0.0	84.9 ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0
	20	81.1	84.6 ±0.0	84.8 ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0	**85.0** ±0.0
	30	59.6	83.5 ±0.1	84.4 ±0.0	84.8 ±0.0	**84.9** ±0.0	**84.9** ±0.0	**84.9** ±0.0
	40	5.2	81.5 ±0.1	83.8 ±0.0	84.5 ±0.0	**84.7** ±0.0	**84.7** ±0.0	**84.7** ±0.0
	50	0.6	78.7 ±0.2	82.5 ±0.1	83.7 ±0.0	84.1 ±0.0	**84.2** ±0.1	83.9 ±0.0
	60	0.1	73.6 ±0.2	79.9 ±0.0	82.3 ±0.0	**83.1** ±0.0	**83.1** ±0.1	80.6 ±0.2
	70	0.1	64.8 ±0.8	74.0 ±0.0	79.6 ±0.1	80.8 ±0.0	80.9 ±0.0	**81.4** ±0.0
	80	0.1	52.9 ±1.1	58.7 ±0.2	73.7 ±0.1	75.4 ±0.0	75.6 ±0.0	**77.8** ±0.0
	90	0.2	25.4 ±0.9	32.3 ±0.5	58.1 ±0.1	59.6 ±0.1	57.0 ±0.3	**66.0** ±0.1

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
ConvNeXt-T								
	0	**Teacher: 84.0**						
	10	38.0	76.8 ±1.0	82.9 ±0.0	83.2 ±0.0	83.3 ±0.0	**83.5** ±0.0	**83.5** ±0.0
	20	4.6	63.6 ±0.9	79.8 ±0.1	81.1 ±0.0	81.8 ±0.0	81.8 ±0.1	**82.3** ±0.0
	30	0.4	35.3 ±2.8	71.9 ±0.1	77.2 ±0.1	78.6 ±0.0	78.9 ±0.1	**79.9** ±0.0
	40	0.1	6.6 ±0.4	55.1 ±0.3	69.5 ±0.1	72.5 ±0.2	73.1 ±0.0	**75.7** ±0.1
	50	0.1	1.0 ±0.1	32.2 ±0.5	57.4 ±0.2	60.5 ±0.0	62.3 ±0.2	**67.9** ±0.1
	60	0.1	0.3 ±0.0	11.0 ±0.1	39.3 ±0.4	41.4 ±0.3	42.1 ±0.5	**53.1** ±0.2
	70	0.1	0.3 ±0.0	1.7 ±0.1	20.4 ±0.4	19.6 ±0.5	17.5 ±0.8	**29.6** ±0.2
	80	0.1	0.1 ±0.0	0.3 ±0.0	5.1 ±0.1	4.0 ±0.1	3.2 ±0.1	**6.9** ±0.2
	90	0.1	0.1 ±0.0	0.2 ±0.0	0.3 ±0.0	0.4 ±0.0	0.2 ±0.0	**0.5** ±0.0
ConvNeXt-S								
	0	**Teacher: 84.9**						
	10	28.0	80.0 ±0.0	83.8 ±0.0	84.2 ±0.0	84.5 ±0.0	84.5 ±0.0	**84.6** ±0.0
	20	3.1	63.8 ±1.2	80.2 ±0.0	82.5 ±0.0	83.1 ±0.0	83.3 ±0.0	**83.7** ±0.1
	30	0.4	25.8 ±0.8	72.1 ±0.1	79.1 ±0.0	80.0 ±0.0	80.6 ±0.0	**82.1** ±0.0
	40	0.1	4.0 ±0.5	55.4 ±0.4	73.1 ±0.1	74.4 ±0.1	75.4 ±0.1	**78.8** ±0.1
	50	0.1	0.8 ±0.2	31.8 ±0.3	62.7 ±0.1	63.3 ±0.1	64.7 ±0.2	**72.7** ±0.2
	60	0.1	0.3 ±0.1	8.5 ±0.4	46.8 ±0.4	45.5 ±0.2	46.4 ±0.2	**60.2** ±0.2
	70	0.1	0.2 ±0.1	1.4 ±0.1	27.0 ±0.1	24.5 ±0.3	24.7 ±0.2	**38.3** ±0.3
	80	0.1	0.1 ±0.0	0.3 ±0.0	7.0 ±0.2	5.6 ±0.2	5.1 ±0.1	**13.0** ±0.3
	90	0.1	0.1 ±0.0	0.1 ±0.0	0.5 ±0.0	0.6 ±0.0	0.5 ±0.0	**0.9** ±0.0
ConvNeXt-B								
	0	**Teacher: 85.5**						
	10	56.2	83.7 ±0.2	84.9 ±0.0	**85.3** ±0.0	**85.3** ±0.0	**85.3** ±0.0	**85.3** ±0.0
	20	1.0	79.5 ±0.1	83.2 ±0.0	84.3 ±0.0	**84.7** ±0.0	**84.7** ±0.0	**84.7** ±0.0
	30	0.2	70.6 ±0.2	79.7 ±0.0	82.7 ±0.1	83.3 ±0.0	83.5 ±0.0	**83.7** ±0.0
	40	0.1	36.0 ±0.8	72.6 ±0.1	79.8 ±0.0	80.7 ±0.1	80.9 ±0.1	**81.6** ±0.0
	50	0.1	2.9 ±0.7	57.8 ±0.5	74.7 ±0.1	75.4 ±0.1	75.8 ±0.1	**77.8** ±0.0
	60	0.1	0.6 ±0.2	34.0 ±0.4	66.0 ±0.1	63.7 ±0.1	64.3 ±0.1	**69.8** ±0.1
	70	0.1	0.3 ±0.1	9.1 ±0.2	50.1 ±0.0	43.4 ±0.4	41.5 ±0.6	**52.8** ±0.4
	80	0.1	0.1 ±0.0	1.2 ±0.1	**25.7** ±0.1	18.0 ±0.2	14.6 ±0.4	22.7 ±0.4
	90	0.1	0.1 ±0.0	0.5 ±0.0	1.7 ±0.1	1.5 ±0.1	0.5 ±0.0	**1.8** ±0.1

M LANGUAGE MODELLING

We keep the one-shot protocol fixed and vary the MLP sparsity on Qwen2.5-1.5B; Table M.1 reports WikiText-2 perplexity.

Table M.1: WikiText-2 perplexity (↓) after one-shot MLP compression, no fine-tuning. Mean ± s.d. over three calibration draws, omitted where the draws do not separate at the printed precision, that is every MAG cell, whose three draws are bit-identical because MAG reads no calibration. Best per row in **bold**, second best underlined; entries above 10^4 are collapsed models. Dense , competing pruning methods , and our method .

Model	Sp. (%)	Baselines						Ours
		MAG	Wanda	VBP	DVBP	CORP	GRAIL	CARVE
Qwen2.5-1.5B								
	0	**Dense: 9.26**						
	10	245	10.37 ±0.03	30.87 ±1.15	15.74 ±0.14	9.86 ±0.01	9.96 ±0.02	**9.50**
	20	2364	12.16 ±0.03	65.50 ±3.83	38.96 ±1.67	10.78 ±0.03	11.04 ±0.02	**10.02** ±0.02
	30	6093	14.73 ±0.03	123 ±7	65.91 ±4.67	11.95 ±0.02	12.34 ±0.03	**10.93** ±0.02
	40	$>10^4$	18.87 ±0.23	275 ±29	133 ±16	13.51 ±0.02	14.04 ±0.09	**12.55** ±0.02
	50	$>10^4$	29.30 ±0.96	1245 ±123	276 ±99	15.69 ±0.08	16.32 ±0.02	**15.47** ±0.04
	60	$>10^4$	73.91 ±3.44	6186 ±590	936 ±138	**19.99** ±0.43	21.00 ±0.56	21.44 ±0.07
	70	$>10^4$	264 ±12	$>10^4$	7165 ±765	35.88 ±2.18	35.97 ±1.00	**35.64** ±0.22
	80	$>10^4$	2942 ±678	$>10^4$	9903 ±2171	143 ±11	206 ±55	**109** ±2
	90	$>10^4$	$>10^4$	$>10^4$	$>10^4$	1351 ±87	3768 ±1065	**1117** ±64

CARVE has the lowest perplexity at eight of nine sparsities and remains below twice the dense perplexity through 50% sparsity; the one exception is 60%, where CORP leads by 1.45 perplexity.

N ALLOCATION ON A DEEPER TRANSFORMER

In the main paper, Figure 2 studies how the pruning budget is distributed across the blocks of DeiT-Base at 70% MLP sparsity. Here, we repeat the same analysis on the deeper DeiT-III-Huge model, which contains 32 Transformer blocks. This experiment helps us understand whether the same allocation effects remain when the network is much deeper.

Elimination scores strongly depend on depth. Figure N.2 shows how the cost of removing a neuron changes as pruning progresses in each block. The curves are very different across depth. For example, after pruning 50% of every block, the score ranges from 1.6×10^{-3} in block 1 to 2.5×10^{-1} in block 23, a factor of about 150. The difference is even larger at some other pruning levels: the ratio between the largest and smallest block scores is about 351 at 30% sparsity and 95 at 70% sparsity.

This large-scale difference explains why directly comparing raw local scores across blocks can be problematic. CARVE_{G} uses the raw CARVE score without cross-block normalisation. It therefore removes neurons until all blocks reach approximately the same raw score threshold. This can be seen in Figure N.2, where the CARVE_{G} markers approximately follow a horizontal line. Blocks with consistently small scores are consequently pruned much more aggressively, in some cases up to 99%.

This allocation has a large effect on the final model. At 70% MLP sparsity, CARVE reaches 81.8% Top-1 accuracy, compared with 74.4% for CARVE_{G}, 76.6% for DVBP, and 74.3% for VBP (Table J.1). These results show that a good local pruning score is not sufficient by itself: how pruning is distributed across blocks also becomes important in deep networks.

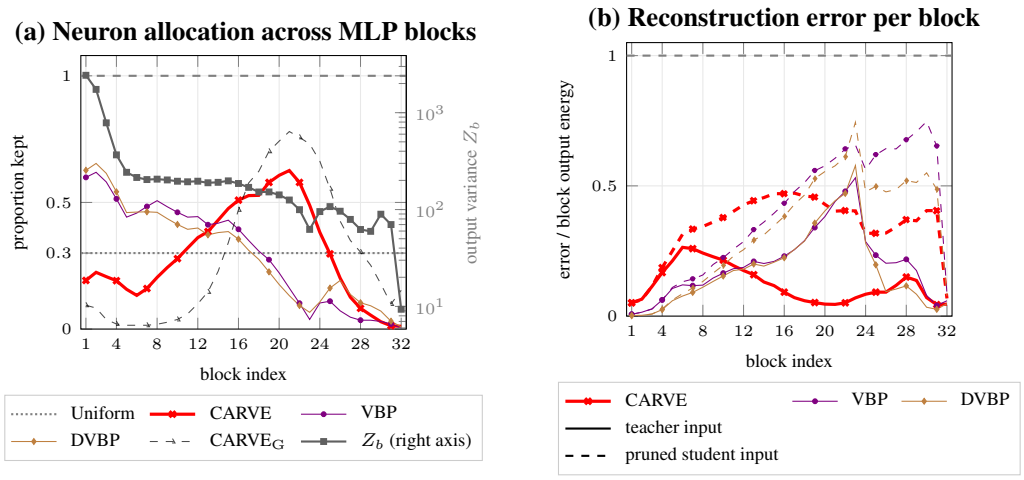

Figure N.1: DeiT-III-Huge on ImageNet-1k at 70% MLP sparsity. **(a)** Fraction of the 5120 hidden neurons kept in each of the 32 MLP blocks for different allocation strategies. The right axis shows the dense MLP output variance Z_b. **(b)** Reconstruction error of each block, normalised by its dense output energy. Solid lines use the clean teacher input, while dashed lines use the effective input produced by the pruned network.

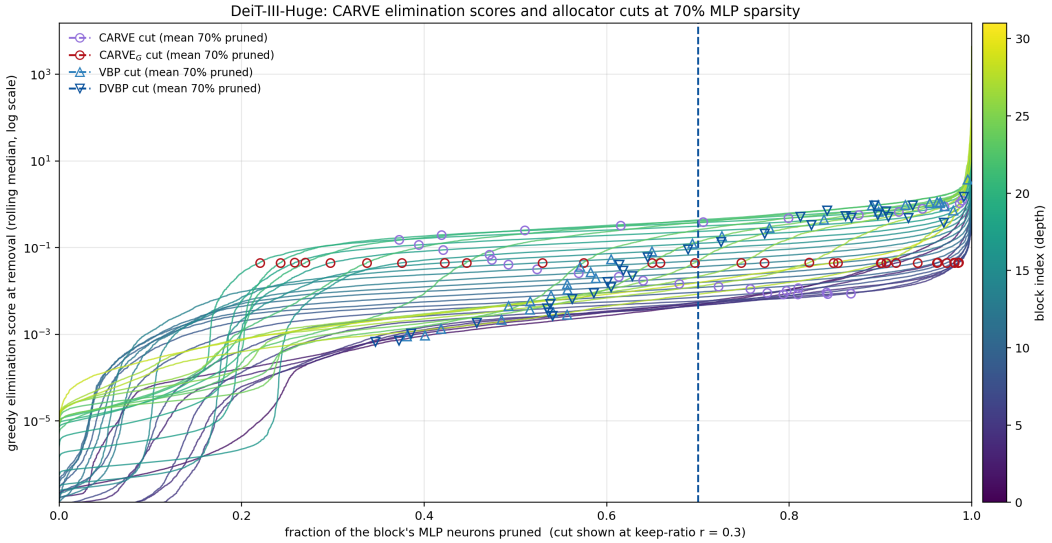

Figure N.2: DeiT-III-Huge on ImageNet-1k. For each block, we show the CARVE elimination score $\|w_i^t\|_2^2/\Omega_{ii}^t$ from equation 14 when each neuron is removed, as a function of the fraction of neurons already pruned. Each curve corresponds to one block and uses a rolling median over 1% of the block. The y-axis is logarithmic and colour indicates depth. Markers show the final pruning level selected by each allocator at 70% global MLP sparsity. The dashed vertical line corresponds to a uniform 70% sparsity in every block.

O Computational cost

Algorithmic cost. For each block b, CARVE first estimates the covariance of the hidden activations from N calibration tokens. For a hidden width H_b, this requires $O(NH_b^2)$ operations. The elimination stage starts by inverting $\Sigma_b + \lambda_b I$, which costs $O(H_b^3)$. When k neurons remain, one elimination step updates the compensated weights and the precision matrix in $O(kD_b + k^2)$ operations. Summing these updates over all pruning steps gives a total cost of $O(H_b^3 + H_b^2 D_b)$ per block, in addition to the calibration cost. The memory cost is $O(H_b^2 + H_b D_b)$, mainly to store the

precision matrix and the compensated output weights. Importantly, the covariance matrix is inverted only once; all later pruning steps use the rank-one updates of Section 3.1.

Tables O.1 and O.2 report the number of parameters, MACs, and peak GPU memory after pruning. MACs are measured with `fvcore` for one 224×224 image; for the Transformers, we add the attention products $QK^\top$ and AV, which `fvcore` does not count. Peak memory is the maximum CUDA memory allocated during a forward pass with a batch of 64 images in bfloat16 on one NVIDIA B200.

Table O.1: Parameters, MACs and peak memory of the Transformer backbones compressed by CARVE, at 30, 50 and 70% MLP sparsity.

	Params (M)				MACs (G)				Peak memory (MiB)			
Backbone	Dense	30%	50%	70%	Dense	30%	50%	70%	Dense	30%	50%	70%
DeiT-Tiny	5.7	4.7	3.9	3.2	1.26	1.05	0.91	0.77	173	162	154	140
DeiT-Small	22.1	17.8	15.0	12.1	4.61	3.77	3.21	2.66	406	371	341	303
DeiT-Base	86.6	69.6	58.2	46.9	17.58	14.24	12.01	9.77	1168	1062	952	861
ViT-B/16	86.6	69.6	58.2	46.9	17.58	14.24	12.01	9.77	1168	1043	966	886
DeiT-III-Large	304.4	243.9	203.7	163.4	61.60	49.71	41.77	33.84	3343	2885	2655	2430
DeiT-III-Huge	632.1	506.2	422.3	338.4	167.40	135.06	113.51	91.95	6831	5859	5391	4912

Table O.2: Parameters, MACs and peak memory of the ConvNeXt backbones for each allocation, at 30, 50 and 70% MLP sparsity.

		Uniform			CARVE			VBP			DVBP		
Backbone	**Sp. (%)**	Params	MACs	Mem.	Params	MACs	Mem.	Params	MACs	Mem.	Params	MACs	Mem.
ConvNeXt-T	30	20.8	3.22	665	20.8	3.44	757	18.4	3.63	733	18.4	3.63	733
	50	15.6	2.39	592	15.5	2.75	738	12.6	2.97	705	12.7	2.96	705
	70	10.4	1.56	530	10.2	1.99	704	7.9	2.13	666	7.9	2.14	667
ConvNeXt-S	30	36.1	6.21	810	37.0	6.33	908	33.9	6.59	883	33.9	6.59	882
	50	26.6	4.54	725	27.8	4.78	871	23.3	5.13	836	23.6	5.12	838
	70	17.2	2.87	638	18.0	3.28	827	14.6	3.45	762	14.7	3.47	766
ConvNeXt-B	30	63.4	10.94	1196	63.9	11.20	1318	59.4	11.58	1303	59.5	11.58	1305
	50	46.7	7.98	1052	47.3	8.43	1254	41.2	8.93	1231	41.7	8.90	1232
	70	29.9	5.02	944	30.1	5.70	1188	25.5	5.98	1149	25.8	5.99	1151

Cost on ConvNeXt. For the Transformer backbones above, all MLP blocks have the same embedding dimension, so removing the same total number of hidden neurons gives almost the same parameter and MAC reduction independently of their allocation. ConvNeXt is different because its width and spatial resolution change across stages. As a result, two neurons from different stages do not have the same cost. A neuron in an early stage has fewer parameters but is evaluated on a large feature map, and therefore has a relatively high computational cost. A neuron in a later stage has more parameters but is evaluated at a lower spatial resolution. Consequently, two pruning strategies with the same overall MLP sparsity can produce different numbers of parameters, MACs, and different peak memory.

Table O.2 illustrates this effect. A uniform allocation gives the lowest MAC count, while VBP and DVBP remove more neurons from the later stages and therefore obtain the smallest parameter counts. CARVE gives an intermediate trade-off. At 70% sparsity, its number of parameters is within 5% of the uniform allocation. Its MAC count is 13.5%–27.6% higher than uniform, but 4.7%–6.6% lower than VBP.

The main cost of the CARVE allocation is memory. Because it keeps more neurons in the first stages, where feature maps are larger, its peak memory is 26%–33% higher than the uniform allocation at 70% sparsity. This is an important difference from the Transformer experiments: on ConvNeXt, comparisons at the same MLP sparsity do *not* correspond exactly to comparisons at the same number of parameters, MACs, or peak memory. The accuracy results in Section 4 should therefore be interpreted as comparisons under the same number of removed MLP neurons.

P RESULTS ON THE CIFAR-100 DATASET

We repeat the one-shot protocol on CIFAR-100 teachers obtained by fine-tuning the public pretrained backbones used elsewhere in the paper. This experiment tests whether an allocation derived from activation geometry remains effective under a change in image distribution and label space, rather than whether the pretrained checkpoints themselves transfer. Every calibration statistic used by CARVE is an expectation over the input distribution, so a teacher fine-tuned on a different distribution induces genuinely different statistics rather than a simple rescaling of those measured on ImageNet. The three ConvNeXt backbones are evaluated on ImageNet-1k only, so the CIFAR-100 grid covers the six Transformer backbones.

The comparison follows the same protocol as in the main paper, and it isolates the effects of selection and allocation. Table P.1 uses the per-block sparsity schedules defined by each method in its original formulation, whereas Table P.2 enforces the same sparsity in every block so that only selection and compensation vary. Each table is continued on the following page.

CARVE achieves the best performance in 48 of the 54 model-sparsity settings under its own allocation, while CARVE_{L} does so in 46 of 54 settings under uniform sparsity. Most of these losses occur at 50% sparsity or below, where all compensated methods are within a few tenths of a point of one another and of the teacher. The margin over the best baseline then increases monotonically with sparsity, from $+0.1$ points at 10% to $+6.0$ at 90%. The ordering of the baselines is consistent with ImageNet-1k: MAG collapses first, followed by SNIP, while the ridge-compensated methods remain competitive over a wider sparsity range. Overall, the CIFAR-100 grid is more compressed: performance remains close to the teacher until roughly half of the hidden units are removed, which is why the main results are reported on ImageNet-1k.

Table P.1: Top-1 (%) on CIFAR-100 across six vision backbones and nine MLP sparsities, one-shot and without fine-tuning. Mean ± standard deviation over three calibration seeds; MAG is deterministic and is run once. Best result in **bold**, second best underlined. Teacher, competing pruning methods, and our method.

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
DeiT-Tiny								
	0	Teacher: 84.5						
	10	76.0	80.8 ±0.2	83.6 ±0.1	84.1 ±0.0	84.1 ±0.0	84.2 ±0.0	**84.5** ±0.0
	20	49.6	74.6 ±0.7	82.1 ±0.1	82.7 ±0.1	83.5 ±0.1	83.5 ±0.0	**83.7** ±0.1
	30	15.2	61.7 ±0.2	79.8 ±0.0	81.0 ±0.1	82.0 ±0.0	**82.3** ±0.0	82.2 ±0.1
	40	7.2	46.0 ±3.1	76.1 ±0.1	78.8 ±0.0	**79.6** ±0.0	**79.6** ±0.0	79.5 ±0.1
	50	3.2	25.1 ±2.2	67.8 ±0.1	73.8 ±0.1	74.9 ±0.1	75.6 ±0.1	**76.6** ±0.1
	60	2.6	9.4 ±1.6	54.4 ±0.1	68.2 ±0.1	67.6 ±0.0	68.8 ±0.1	**72.1** ±0.1
	70	1.1	4.1 ±1.0	29.1 ±0.0	58.6 ±0.2	56.3 ±0.1	58.0 ±0.2	**64.4** ±0.3
	80	1.1	2.4 ±0.6	7.8 ±0.2	37.3 ±0.7	40.1 ±0.3	41.3 ±0.2	**50.7** ±0.5
	90	0.9	1.2 ±0.2	3.4 ±0.2	10.6 ±0.2	16.5 ±0.2	13.1 ±0.2	**22.5** ±0.4
DeiT-Small								
	0	Teacher: 87.5						
	10	81.0	86.1 ±0.1	86.8 ±0.0	87.0 ±0.0	87.0 ±0.0	86.8 ±0.1	**87.4** ±0.0
	20	64.8	83.0 ±0.3	85.4 ±0.0	86.6 ±0.0	**86.9** ±0.1	85.0 ±0.1	86.8 ±0.0
	30	26.7	74.6 ±0.5	84.6 ±0.1	86.1 ±0.0	86.3 ±0.0	85.4 ±0.1	**86.4** ±0.0
	40	13.9	58.9 ±1.1	83.3 ±0.1	85.7 ±0.0	85.4 ±0.1	85.3 ±0.1	**85.8** ±0.0
	50	4.9	40.4 ±1.1	81.1 ±0.0	**84.9** ±0.0	83.5 ±0.0	84.0 ±0.0	84.5 ±0.1
	60	1.6	20.3 ±0.8	77.5 ±0.1	82.0 ±0.1	81.2 ±0.2	81.4 ±0.0	**83.1** ±0.0
	70	1.0	8.6 ±0.3	66.7 ±0.2	78.5 ±0.2	77.1 ±0.3	77.4 ±0.2	**80.2** ±0.1
	80	1.1	4.8 ±0.4	37.6 ±0.4	69.8 ±0.4	67.6 ±0.1	69.5 ±0.3	**74.0** ±0.1
	90	1.6	2.0 ±0.4	13.1 ±0.2	42.6 ±0.2	46.4 ±0.2	47.8 ±0.2	**58.9** ±0.1
DeiT-Base								
	0	Teacher: 90.3						
	10	75.7	89.4 ±0.1	**90.3** ±0.0	**90.3** ±0.0	90.2 ±0.0	89.9 ±0.1	**90.3** ±0.0
	20	27.7	86.3 ±0.2	89.5 ±0.1	**90.3** ±0.0	90.1 ±0.0	89.8 ±0.0	**90.3** ±0.0
	30	3.7	81.0 ±0.7	88.2 ±0.0	90.0 ±0.0	89.9 ±0.0	90.0 ±0.1	**90.3** ±0.0
	40	1.1	73.2 ±0.9	86.6 ±0.0	89.6 ±0.1	89.4 ±0.1	89.4 ±0.1	**90.1** ±0.0
	50	0.8	62.7 ±1.0	82.8 ±0.1	88.6 ±0.1	88.5 ±0.1	88.7 ±0.0	**89.7** ±0.0
	60	1.1	44.8 ±0.7	63.1 ±0.2	86.1 ±0.1	86.8 ±0.1	87.3 ±0.0	**88.9** ±0.0
	70	1.1	21.9 ±0.8	18.0 ±0.3	81.0 ±0.0	83.9 ±0.0	84.6 ±0.1	**86.9** ±0.1
	80	1.6	7.9 ±2.1	5.3 ±0.2	61.3 ±0.4	77.7 ±0.3	79.3 ±0.2	**83.1** ±0.1
	90	1.0	1.9 ±0.5	3.5 ±0.1	9.3 ±0.2	61.0 ±0.1	63.8 ±0.3	**72.0** ±0.3

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP	DVBP	CORP	GRAIL	CARVE
ViT-B/16								
	0	**Teacher: 93.6**						
	10	77.6	92.5 ±0.0	93.2 ±0.0	**93.7** ±0.0	93.5 ±0.0	93.5 ±0.0	93.6 ±0.0
	20	45.5	90.2 ±0.2	92.4 ±0.0	93.6 ±0.0	93.3 ±0.1	93.3 ±0.0	**93.7** ±0.0
	30	7.1	85.8 ±0.3	91.3 ±0.1	93.4 ±0.0	93.0 ±0.0	93.1 ±0.0	**93.5** ±0.0
	40	1.2	75.8 ±1.1	89.9 ±0.1	92.7 ±0.0	92.5 ±0.1	92.8 ±0.0	**93.3** ±0.0
	50	0.9	48.9 ±2.1	86.9 ±0.1	91.9 ±0.0	91.5 ±0.0	91.8 ±0.1	**92.7** ±0.0
	60	1.0	17.1 ±1.6	73.5 ±0.2	89.9 ±0.1	89.9 ±0.1	90.4 ±0.0	**91.8** ±0.0
	70	1.0	3.3 ±0.5	36.9 ±0.2	86.5 ±0.1	87.5 ±0.2	87.7 ±0.2	**89.8** ±0.1
	80	1.0	1.5 ±0.4	11.9 ±0.3	75.3 ±0.3	82.3 ±0.1	82.6 ±0.1	**85.5** ±0.1
	90	1.0	0.9 ±0.1	3.4 ±0.1	16.6 ±0.2	66.4 ±0.1	66.9 ±0.0	**70.5** ±0.4
DeiT-III-Large								
	0	**Teacher: 91.8**						
	10	85.7	91.6 ±0.0	91.7 ±0.0	91.8 ±0.0	91.8 ±0.0	**91.9** ±0.0	**91.9** ±0.0
	20	69.8	90.5 ±0.1	91.5 ±0.0	91.7 ±0.0	91.7 ±0.0	**91.8** ±0.0	**91.8** ±0.0
	30	25.3	88.2 ±0.1	90.7 ±0.0	91.3 ±0.0	91.4 ±0.0	91.5 ±0.1	**91.6** ±0.0
	40	9.4	82.6 ±0.2	88.3 ±0.1	90.6 ±0.1	**91.2** ±0.0	91.1 ±0.0	**91.2** ±0.1
	50	6.5	71.7 ±1.2	85.2 ±0.1	89.7 ±0.2	90.6 ±0.1	90.2 ±0.0	**90.8** ±0.0
	60	1.6	60.9 ±1.6	81.7 ±0.1	88.4 ±0.1	89.5 ±0.0	89.3 ±0.1	**90.5** ±0.1
	70	1.4	49.1 ±1.7	78.1 ±0.1	86.6 ±0.0	87.9 ±0.1	87.7 ±0.0	**89.3** ±0.1
	80	1.0	31.9 ±2.8	67.6 ±0.2	84.0 ±0.1	83.8 ±0.1	83.7 ±0.0	**86.6** ±0.1
	90	1.1	5.0 ±0.8	28.6 ±0.3	73.0 ±0.2	71.9 ±0.2	69.3 ±0.1	**76.2** ±0.2
DeiT-III-Huge								
	0	**Teacher: 91.9**						
	10	84.9	91.7 ±0.1	91.7 ±0.0	91.8 ±0.0	**91.9** ±0.0	91.8 ±0.0	91.8 ±0.0
	20	72.0	91.4 ±0.0	91.1 ±0.1	91.4 ±0.0	**91.8** ±0.0	91.7 ±0.0	**91.8** ±0.0
	30	41.6	90.3 ±0.1	90.2 ±0.1	90.9 ±0.1	91.5 ±0.0	91.4 ±0.0	**91.6** ±0.0
	40	14.0	88.1 ±0.2	88.9 ±0.1	89.8 ±0.1	91.2 ±0.0	**91.3** ±0.0	**91.3** ±0.0
	50	5.0	83.3 ±0.2	87.6 ±0.0	88.9 ±0.0	90.5 ±0.0	90.3 ±0.0	**90.7** ±0.1
	60	2.8	73.9 ±0.5	85.5 ±0.1	88.2 ±0.1	89.1 ±0.0	89.1 ±0.0	**89.7** ±0.1
	70	1.5	60.8 ±0.5	81.4 ±0.1	87.3 ±0.1	87.2 ±0.1	87.3 ±0.1	**88.6** ±0.1
	80	1.3	46.6 ±0.8	72.4 ±0.2	84.9 ±0.0	84.5 ±0.1	85.2 ±0.1	**86.7** ±0.1
	90	1.8	6.7 ±1.3	42.2 ±0.3	76.1 ±0.1	76.2 ±0.1	76.1 ±0.2	**80.1** ±0.1

Table P.2: Top-1 (%) on CIFAR-100 at uniform per-block sparsity, across the six vision backbones and nine MLP sparsities. Only neuron selection and compensation vary. Mean ± standard deviation over three calibration seeds; MAG is deterministic and is run once. Best result in **bold**, second best underlined. Teacher , competing pruning methods , and our method .

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
DeiT-Tiny								
	0	**Teacher: 84.5**						
	10	76.0	80.8 ±0.2	83.7 ±0.1	83.9 ±0.1	84.1 ±0.0	**84.2** ±0.0	**84.2** ±0.1
	20	49.6	74.6 ±0.7	81.8 ±0.1	82.8 ±0.1	**83.5** ±0.1	**83.5** ±0.0	**83.5** ±0.1
	30	15.2	61.7 ±0.2	78.8 ±0.1	80.5 ±0.1	82.0 ±0.0	82.3 ±0.0	**82.4** ±0.1
	40	7.2	46.0 ±3.1	73.0 ±0.1	77.3 ±0.2	79.6 ±0.0	79.6 ±0.0	**80.0** ±0.1
	50	3.2	25.1 ±2.2	62.6 ±0.2	72.7 ±0.1	74.9 ±0.1	75.6 ±0.1	**76.9** ±0.2
	60	2.6	9.4 ±1.6	46.9 ±0.1	66.6 ±0.2	67.6 ±0.0	68.8 ±0.1	**70.6** ±0.1
	70	1.1	4.1 ±1.0	22.0 ±0.8	51.0 ±0.3	56.3 ±0.1	58.0 ±0.2	**62.1** ±0.1
	80	1.1	2.4 ±0.6	11.5 ±0.3	35.6 ±0.2	40.1 ±0.3	41.3 ±0.2	**47.3** ±0.0
	90	0.9	1.2 ±0.2	4.6 ±0.2	14.6 ±0.2	16.5 ±0.2	13.1 ±0.2	**23.2** ±0.4
DeiT-Small								
	0	**Teacher: 87.5**						
	10	81.0	86.1 ±0.1	86.9 ±0.0	**87.1** ±0.0	87.0 ±0.0	86.8 ±0.1	**87.1** ±0.1
	20	64.8	83.0 ±0.3	86.2 ±0.1	86.8 ±0.0	86.9 ±0.1	85.0 ±0.1	**87.1** ±0.1
	30	26.7	74.6 ±0.5	85.2 ±0.1	86.0 ±0.0	86.3 ±0.0	85.4 ±0.1	**86.6** ±0.0
	40	13.9	58.9 ±1.1	82.8 ±0.1	84.8 ±0.2	85.4 ±0.1	85.3 ±0.1	**85.7** ±0.1
	50	4.9	40.4 ±1.1	79.0 ±0.1	82.5 ±0.2	83.5 ±0.0	84.0 ±0.0	**84.5** ±0.1
	60	1.6	20.3 ±0.8	73.4 ±0.2	80.8 ±0.1	81.2 ±0.2	81.4 ±0.0	**82.0** ±0.0
	70	1.0	8.6 ±0.3	61.5 ±0.3	76.3 ±0.1	77.1 ±0.3	77.4 ±0.2	**78.3** ±0.1
	80	1.1	4.8 ±0.4	42.5 ±0.3	65.8 ±0.1	67.6 ±0.1	69.5 ±0.3	**72.5** ±0.1
	90	1.6	2.0 ±0.4	14.1 ±0.4	44.5 ±0.3	46.4 ±0.2	47.8 ±0.2	**55.2** ±0.2
DeiT-Base								
	0	**Teacher: 90.3**						
	10	75.7	89.4 ±0.1	89.4 ±0.1	89.5 ±0.0	**90.2** ±0.0	89.9 ±0.1	**90.2** ±0.0
	20	27.7	86.3 ±0.2	89.8 ±0.1	**90.1** ±0.1	**90.1** ±0.0	89.8 ±0.0	**90.1** ±0.0
	30	3.7	81.0 ±0.7	88.7 ±0.1	89.7 ±0.1	89.9 ±0.0	**90.0** ±0.1	**90.0** ±0.0
	40	1.1	73.2 ±0.9	87.2 ±0.2	88.8 ±0.1	89.4 ±0.1	89.4 ±0.1	**89.6** ±0.1
	50	0.8	62.7 ±1.0	83.9 ±0.1	87.6 ±0.0	88.5 ±0.1	88.7 ±0.0	**89.1** ±0.0
	60	1.1	44.8 ±0.7	78.2 ±0.1	85.9 ±0.0	86.8 ±0.1	87.3 ±0.0	**87.7** ±0.1
	70	1.1	21.9 ±0.8	67.0 ±0.2	83.1 ±0.1	83.9 ±0.0	84.6 ±0.1	**85.4** ±0.1
	80	1.6	7.9 ±2.1	44.5 ±0.3	77.0 ±0.3	77.7 ±0.3	79.3 ±0.2	**81.2** ±0.2
	90	1.0	1.9 ±0.5	19.6 ±0.4	57.1 ±0.4	61.0 ±0.1	63.8 ±0.3	**71.5** ±0.3

Model	Sp. (%)	Baselines						Ours
		MAG	SNIP	VBP_L	DVBP_L	CORP	GRAIL	CARVE_L
ViT-B/16								
	0	**Teacher: 93.6**						
	10	77.6	92.5 ±0.0	93.3 ±0.0	93.5 ±0.0	93.5 ±0.0	93.5 ±0.0	**93.6** ±0.1
	20	45.5	90.2 ±0.2	92.6 ±0.0	93.2 ±0.0	93.3 ±0.1	93.3 ±0.0	**93.4** ±0.0
	30	7.1	85.8 ±0.3	91.8 ±0.1	92.8 ±0.1	93.0 ±0.0	**93.1** ±0.0	93.0 ±0.1
	40	1.2	75.8 ±1.1	89.7 ±0.2	92.0 ±0.1	92.5 ±0.1	**92.8** ±0.0	92.7 ±0.0
	50	0.9	48.9 ±2.1	86.9 ±0.1	90.7 ±0.2	91.5 ±0.0	91.8 ±0.1	**92.3** ±0.1
	60	1.0	17.1 ±1.6	81.5 ±0.2	89.4 ±0.1	89.9 ±0.1	90.4 ±0.0	**90.8** ±0.1
	70	1.0	3.3 ±0.5	68.1 ±0.2	87.2 ±0.1	87.5 ±0.2	87.7 ±0.2	**89.0** ±0.0
	80	1.0	1.5 ±0.4	45.0 ±0.5	82.3 ±0.2	82.3 ±0.1	82.6 ±0.1	**84.6** ±0.2
	90	1.0	0.9 ±0.1	19.2 ±0.3	64.8 ±0.1	66.4 ±0.1	66.9 ±0.0	**72.4** ±0.1
DeiT-III-Large								
	0	**Teacher: 91.8**						
	10	85.7	91.6 ±0.0	91.6 ±0.0	91.7 ±0.0	91.8 ±0.0	**91.9** ±0.0	**91.9** ±0.0
	20	69.8	90.5 ±0.1	90.5 ±0.0	91.5 ±0.0	91.7 ±0.0	**91.8** ±0.0	**91.8** ±0.0
	30	25.3	88.2 ±0.1	89.2 ±0.0	91.0 ±0.0	91.4 ±0.0	**91.5** ±0.1	**91.5** ±0.0
	40	9.4	82.6 ±0.2	87.1 ±0.0	90.6 ±0.0	**91.2** ±0.0	91.1 ±0.0	91.1 ±0.1
	50	6.5	71.7 ±1.2	81.6 ±0.1	89.9 ±0.1	**90.6** ±0.1	90.2 ±0.0	90.5 ±0.0
	60	1.6	60.9 ±1.6	76.7 ±0.1	88.6 ±0.1	89.5 ±0.0	89.3 ±0.1	**89.8** ±0.1
	70	1.4	49.1 ±1.7	72.1 ±0.2	86.9 ±0.1	87.9 ±0.1	87.7 ±0.0	**88.1** ±0.1
	80	1.0	31.9 ±2.8	59.9 ±0.1	83.2 ±0.0	83.8 ±0.1	83.7 ±0.0	**85.4** ±0.1
	90	1.1	5.0 ±0.8	33.0 ±0.2	72.7 ±0.1	71.9 ±0.2	69.3 ±0.1	**77.6** ±0.4
DeiT-III-Huge								
	0	**Teacher: 91.9**						
	10	84.9	91.7 ±0.1	91.7 ±0.0	91.8 ±0.0	**91.9** ±0.0	91.8 ±0.0	91.8 ±0.0
	20	72.0	91.4 ±0.0	91.4 ±0.0	91.7 ±0.0	**91.8** ±0.0	91.7 ±0.0	**91.8** ±0.0
	30	41.6	90.3 ±0.1	90.0 ±0.0	91.1 ±0.0	91.5 ±0.0	91.4 ±0.0	**91.6** ±0.0
	40	14.0	88.1 ±0.2	88.4 ±0.0	90.6 ±0.0	91.2 ±0.0	91.3 ±0.0	**91.5** ±0.0
	50	5.0	83.3 ±0.2	79.3 ±0.1	88.8 ±0.1	90.5 ±0.0	90.3 ±0.0	**90.6** ±0.1
	60	2.8	73.9 ±0.5	66.3 ±0.9	87.6 ±0.1	89.1 ±0.0	89.1 ±0.0	**89.4** ±0.0
	70	1.5	60.8 ±0.5	58.0 ±0.6	86.1 ±0.1	87.2 ±0.1	**87.3** ±0.1	86.3 ±0.2
	80	1.3	46.6 ±0.8	59.3 ±0.7	84.0 ±0.1	84.5 ±0.1	**85.2** ±0.1	83.1 ±0.2
	90	1.8	6.7 ±1.3	42.6 ±0.4	**77.4** ±0.2	76.2 ±0.1	76.1 ±0.2	73.0 ±3.1

Q CALIBRATION SET

We study how sensitive the pruning methods are to the calibration set. We vary two factors: the number of calibration images and the way these images are sampled from ImageNet-1k. We compare a class-balanced subset with a standard uniform random subset. All other settings are kept fixed. MAG is not included in this experiment because it only uses the model weights and does not require calibration data.

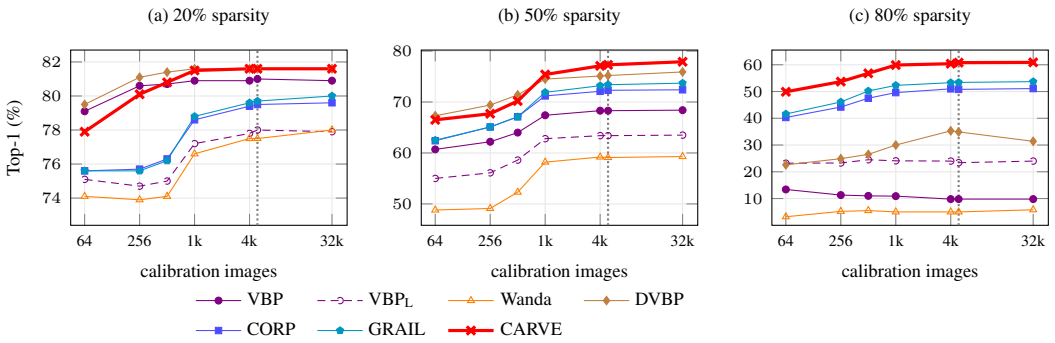

Figure Q.1: DeiT-Base Top-1 accuracy on ImageNet-1k as a function of the number of calibration images, at 20%, 50%, and 80% MLP sparsity. The calibration set is class-balanced. The dotted vertical line indicates the default calibration size of 5,000 images used in the main experiments.

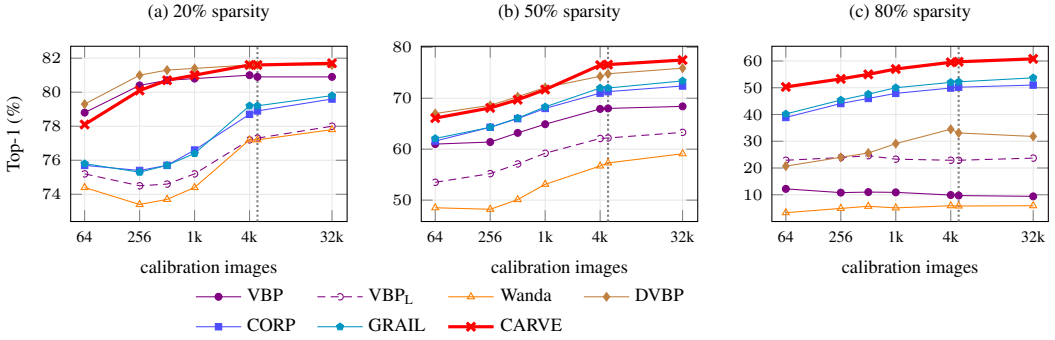

Figure Q.2: Same experiment as Figure Q.1, but using a uniformly sampled calibration set instead of a class-balanced one.

Effect of the calibration set size. Figures Q.1 and Q.2 show a clear trend: very small calibration sets hurt most methods, while performance becomes much more stable once enough images are available. This effect is especially strong below roughly 1,000 images. ImageNet-1k contains 1,000 classes, so a very small calibration set cannot represent the full diversity of the dataset. As a result, the estimated activation statistics become less representative of the model's usual behaviour.

For example, at 50% sparsity with class-balanced sampling (Figure Q.1), reducing the calibration set from 1,024 to 512 images causes a loss of 3.1–5.9 Top-1 points, depending on the method. In comparison, reducing it from 4,096 to 1,024 images causes only 0.6–1.7 points of degradation. This suggests that most of the benefit comes from reaching a sufficiently large and diverse calibration set; beyond this point, adding more images gives only small improvements.

The curves indeed become almost flat between 4,096, 5,000, and 32,768 images. This supports our choice of 5,000 calibration images in the main experiments: it lies in the stable region while remaining much smaller than the full ImageNet training set.

Balanced versus uniform sampling. Class balancing has only a limited effect once the calibration set is large enough. At our default size of 5,000 images, the difference between balanced and

uniform sampling is at most 1.8 Top-1 points. The balanced subset performs at least as well as the uniform subset in 20 of the 21 method–sparsity settings considered here. The only exception is Wanda at 80% sparsity, where uniform sampling is better by 0.8 points.

We therefore use class-balanced calibration in the main experiments. Importantly, class labels are used only to construct the balanced calibration subset; the pruning criterion itself does not use labels.

Impact on the comparison between methods. The calibration size can slightly change the ranking of methods when pruning is mild and the calibration set is very small. At 20% sparsity, DVBP performs better than CARVE for calibration sets smaller than $4{,}096$ images, while the two methods become essentially tied for larger sets. This is also the easiest pruning regime, where most methods remain close to the dense teacher.

The picture is much clearer at stronger compression. At 80% sparsity, CARVE outperforms every baseline for every calibration size and for both sampling strategies, with gains between 6.6 and 10.1 Top-1 points. At 50% sparsity with class-balanced sampling, once the calibration set is large enough to cover the dataset well, CARVE leads the strongest baseline by 0.9–2.1 points. With very small calibration sets, it can trail DVBP by at most 1.7 points.

Overall, the main conclusions of the paper are therefore not sensitive to the exact calibration-set size, provided that the set is large enough to estimate representative activation statistics.

Effect of cross-block allocation. The difference between VBP and VBP_{L} also illustrates the importance of cross-block allocation. VBP_{L} uses the same local VBP criterion but applies a uniform sparsity across blocks, whereas VBP uses its own global allocation strategy.

At 20% and 50% sparsity with class-balanced sampling, the VBP allocator improves accuracy by 3.0–6.1 points over VBP_{L}. However, at 80% sparsity, the same allocation becomes harmful: VBP reaches only about 9.8% Top-1, compared with about 24.0% for VBP_{L}. This again shows that cross-block allocation becomes especially important under aggressive pruning.

DVBP also shows a small non-monotonic behaviour at 80% sparsity. Its best result is obtained with about $4{,}096$ calibration images, and its accuracy decreases by 2.7–3.9 points when increasing the calibration set to $32{,}768$ images. In contrast, CARVE remains stable in this regime.