PRUNING THEN PERFORATING: DENDRITES RECOVER WHAT STRUCTURED PRUNING REMOVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deploying a vision model under a parameter budget pits two techniques against each other: structured pruning cuts parameters by removing capacity, while Perforated Backpropagation adds capacity back through artificial dendrite nodes grown during training. Each is a good mechanism to reach for when searching for a method to optimize accuracy and parameter count, but whether they compose has not been measured. We measure it on half-width ResNet-18 over Oxford-IIIT Pets performing structured pruning based on L2 magnitude and Taylor importance, as well as on a DeepLabV3 segmentation model deployed on an industrial robotics platform performing structured pruning on L2 magnitude only. Against matched zero-dendrite baselines at equal parameter count, perforated models come out ahead at every pruning rate. Composing these methods therefore is optimal compared to either method independently.

1 INTRODUCTION

Shrinking a trained network to fit a parameter budget usually means removing something. Structured pruning deletes channels, and with them the capacity those channels carried (Liu et al., 2017; Fang et al., 2023). Perforated Backpropagation works in the other direction (Brenner & Itti, 2025). It attaches a small number of dendrite nodes to existing neurons and trains them to predict each neuron's remaining error. Brenner & Itti (2025) and Brenner et al. (2025) report accuracy gains, and substantial compression at equal accuracy, across a range of models and tasks. One method removes capacity, the other adds capacity back in a cheap and targeted form, so the obvious question is whether they compose.

Dendrites add parameters, and the perforation pipeline retrains the pruned model before any dendrite exists, so measuring the composition means holding both constant. This paper does that. Every comparison is made at equal parameter count, against a reference taken from the same perforation run at the moment before its first dendrite, which we call the *zero-dendrite baseline*.

We report three studies that differ in pruning schedule, task, and pruning criterion. **Study A** is a controlled benchmark: half-width ResNet-18 on Oxford-IIIT Pets (Parkhi et al., 2012), one-shot structured pruning at nine rates from 10 to 90 percent, six seeds at each rate (Section 4). **Study B** repeats that benchmark with Taylor importance in place of L2 magnitude, changing which channels are removed and how much accuracy the pruned model keeps at a fixed prune rate (Section 5). **Study C** is a production study on a DeepLabV3 segmentation model (Chen et al., 2017) with an Atrous Spatial Pyramid Pooling (ASPP) head and a ResNet-18 backbone (He et al., 2016), deployed on an industrial robotics platform for human detection. It covers successive rounds of structured pruning combined with perforation (Section 6).

Every headline comparison in the experimentation is made against that zero-dendrite baseline. Perforated models sit $+1.15$ points above the zero-dendrite reference over 54 runs, a mean with a 95 percent confidence interval of $[0.72, 1.58]$, and the rate mean is positive at all nine prune rates. This means that adding dendritic parameters with perforation consistently recovers error with a better parameter efficiency than the error penalty of pruning to the *zero-dendrite baseline*.

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.

Contributions.

1. A parameter-matched measurement of structured pruning followed by Perforated Backpropagation in three experimental settings that differ in schedule, task, and criterion, against a zero-dendrite baseline.
2. A split of the parameter-matched gain into a recovery term and a parameter cost term, with the parameter cost small in all three studies.

2 Background and related work

Structured pruning. Structured pruning produces a smaller dense network that runs on any runtime, without the kernel support unstructured sparsity requires. The unit of removal is a dependency group rather than a tensor slice: removing an output channel forces removal of the matching input channels of its consumers, and residual connections couple channels across blocks. The DepGraph formulation (Fang et al., 2023) makes these groups explicit and scores each with an importance criterion, such as the L2 magnitude of its weights or a first-order Taylor estimate of the loss change its removal would cause (Molchanov et al., 2019). Pruning followed by fine-tuning is the standard recipe (Han et al., 2015), extended to channels by criteria such as network slimming (Liu et al., 2017), and the lottery ticket literature (Frankle & Carbin, 2019) reframed it as finding trainable subnetworks. Across this work the architecture is fixed once the mask is chosen, and recovery means re-optimizing the surviving weights. We ask what happens when recovery may add a different kind of unit instead.

Perforated Backpropagation. Biological dendrites are not passive cables; they perform local nonlinear integration before the soma. Perforated Backpropagation takes this as a design cue and, rather than widening layers or adding neurons, adds dendritic nodes to neurons that already exist (Brenner & Itti, 2025). Training alternates two phases, generalizing Cascade-Correlation (Fahlman & Lebiere, 1989) from single-layer networks to deep ones. In the neuron phase, standard backpropagation trains the network until validation performance plateaus. In the dendrite phase, neuron weights are frozen and candidate dendrite nodes are trained to correlate with individual neurons' backpropagated error rather than with the network's output error; the best candidate per neuron is frozen and added to the forward pass, and the cycle repeats until further dendrites stop improving validation performance. Error is never propagated through dendrite nodes, so they sit outside the gradient-descent graph. Two properties matter here. The added capacity is cheap: a dendrite on a linear layer costs about one copy of that layer's weights. And it is targeted: because the dendrite phase fits residual error, a network with little residual error on its training set offers a dendrite little to fit. Brenner & Itti (2025) and Brenner et al. (2025) report accuracy improvements and compression across several domains. Those evaluations apply the method at the model's natural size or without a parameter-matched control; here the models are made capacity-limited deliberately and compared at matched size.

Growing and searching. Dynamic sparse training interleaves removal and growth: SET (Mocanu et al., 2018) regrows connections at random, RigL (Evci et al., 2020) regrows using dense gradients, and Top-KAST (Jayakumar et al., 2020) keeps both passes sparse. These methods grow connections of the same kind at a fixed sparsity, during training, and target unstructured sparsity. Here the grown unit is structurally different, it is added after pruning and fine-tuning have converged, and the parameter count changes, so the question is whether the accuracy-per-parameter frontier moves. Resource-constrained search takes a third route: MorphNet (Gordon et al., 2018) alternates a resource-targeted shrinking regularizer with uniform expansion, and NetAdapt (Yang et al., 2018) reduces layer widths against measured hardware cost. Both search over widths and neither introduces a different kind of unit.

3 Shared protocol and metric

All three studies compare perforated models with unperforated ones at the same or nearly the same parameter count; Studies A and B do so against the interpolated reference curve defined below, which also supports the decomposition. Let n be a model's trainable parameter count. Given a

reference curve R measured at a set of prune rates, $\hat{R}(n)$ denotes linear interpolation of R in $\log_{10} n$. For a perforated model with test accuracy A and n parameters, the *parameter-matched gain* against R is $m_R = A - \hat{R}(n)$, evaluated at the model's own total parameter count, so every parameter a dendrite adds is paid for by the reference.

Budgets and references. A perforation run adds up to three dendrites. *Budget* N is the architecture selected at the best-validation epoch among all epochs with at most N dendrites installed. Budget 0 is therefore the pre-dendrite model, at exactly the pruned parameter count, and reported perforated models are budget 3 unless stated otherwise. In two of the 54 Study A runs the best-validation epoch at budget 3 carries no dendrite, so their raw gain is exactly zero; they are kept, and they lower the pooled gain from $+1.21$ to $+1.15$ points.

The *zero-dendrite reference* Z is the seed-averaged budget-0 test accuracy at each prune rate, placed at that rate's pruned parameter count, so budget 0 lands on Z by construction. Z is defined out to the 10 percent prune rate and extended flat beyond it, where the 10 percent budget models sit.

Isolating the dendrite contribution. The perforation pipeline starts from the converged pruning-only checkpoint, restarts the optimizer and learning-rate schedule, trains again before the first dendrite is added, and restarts them after every restructure. A parameter-matched comparison has to hold that additional training constant, which is what Z does: each run supplies its own reference at the moment before its first dendrite.

The additional training is modest. In Study A the network trains for 20 to 63 epochs (median 34) before the first dendrite phase, against pruning-only fine-tuning of 18 to 59 epochs (median 37.5), so its median is shorter than the fine-tuning it follows.

Decomposition. Against the zero-dendrite reference Z, a perforated run's gain splits into the part already present before any dendrite, $m_{\mathrm{pre}} = A_{\mathrm{pre}} - \hat{Z}(n_{\mathrm{start}})$, and the part added afterwards, $m_Z - m_{\mathrm{pre}}$, where A_{pre} is the pre-dendrite test accuracy and n_{start} the pruned parameter count. Writing the raw gain as $g = A_{\mathrm{final}} - A_{\mathrm{pre}}$, the second term is

$$m_Z - m_{\mathrm{pre}} = g - c, \qquad c = \hat{Z}(n_{\mathrm{final}}) - \hat{Z}(n_{\mathrm{start}}) = s\,(\log_{10} n_{\mathrm{final}} - \log_{10} n_{\mathrm{start}})\,, \quad (1)$$

where s is the local slope of Z between the two parameter counts. The parameter cost c is what the added parameters would have bought by pruning less. The second equality is exact when n_{start} and n_{final} lie on one segment of Z. Against Z the pre-dendrite term vanishes in the seed mean by construction, so the headline parameter-matched gain is the raw gain minus the parameter cost alone.

4 Study A: controlled benchmark on Oxford-IIIT Pets

Setup. Oxford-IIIT Pets (Parkhi et al., 2012) has 37 classes, a trainval split of 3,680 images and a test split of 3,669. Validation is a stratified per-class 80/20 carve-out of trainval (2,944 training and 736 validation images) drawn once with a fixed seed independent of the experiment seed, so every run sees the same partition; the test split is used for reporting only. The model is an ImageNet-pretrained torchvision ResNet-18 (He et al., 2016) with its classifier replaced by a 37-way linear layer, shrunk to width 0.5 by uniform channel removal with the classifier excluded, giving 2,808,389 parameters before pruning. Pruning is one-shot structured channel removal over DepGraph groups with group L2 magnitude importance applied within each group (Fang et al., 2023), at rates of 10 to 90 percent, leaving between 2,264,295 and 27,822 parameters (Appendix A). Channels are physically removed and a verification pass rejects any masked parameters, so reported counts are those of the executed network. Every pruned model is fine-tuned to convergence with patience-based early stopping and no epoch cap (Appendix D). Dendrites attach to the final classifier only, the network's single linear layer, with at most three dendrites and three attempts per dendrite, using the fine-tuning optimizer and schedule. One dendrite costs `in_features` $\times\, 37 + 37$ parameters, from 8,547 at 10 percent to 962 at 90 percent, and budget-3 models add 0.8 to 10.4 percent parameters. The population is 54 runs: nine prune rates crossed with six seeds.

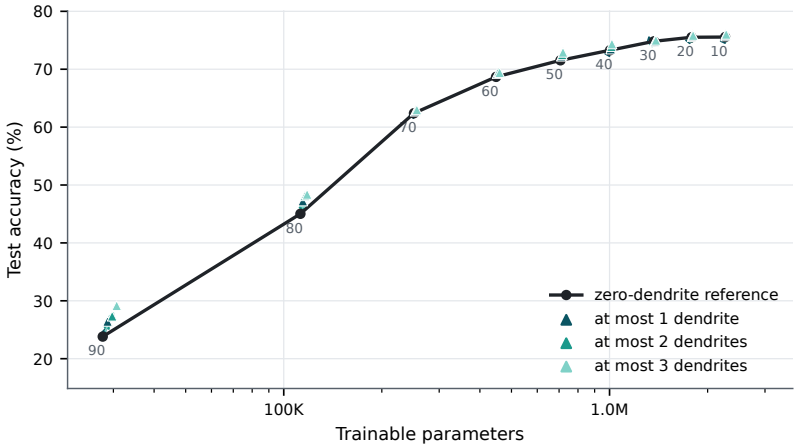

Figure 1: Study A, L2 magnitude. The zero-dendrite reference (solid, circles) against trainable parameters on a log scale; labels give the prune rate. Triangles mark the mean model accuracy achieved with at most one, two or three dendrites, at its own parameter count. Axis ranges, styles and legend match Figure 2.

4.1 Against the zero-dendrite reference

Pooled over all 54 runs, perforated models sit $+1.15$ points $[0.72, 1.58]$ above the zero-dendrite reference at their own parameter count, with 43 of 54 runs positive ($p = 1.5 \times 10^{-6}$). The rate mean is positive at all nine prune rates, and the effect is concentrated at the heavy rates: it is largest at 80 and 90 percent, where pruning has removed the most capacity, averaging $+3.08$ points there against $+0.60$ over rates 10 to 70. It is not monotone in the prune rate below 80 percent, where the rate means range from $+0.14$ at 30 percent to $+1.30$ at 50 percent; the Spearman rank correlation between prune rate and gain across the nine rate means is $\rho = 0.68$ ($p = 0.042$). Per-rate gains, intervals and tests are in Table 1. Restricted to rates 40 through 90 the gain is $+1.57$ $[0.99, 2.15]$.

Table 1: Study A, parameter-matched gain against the zero-dendrite reference at budget 3, by prune rate. Confidence intervals are 95 percent t-intervals; p is a two-sided one-sample t-test. Bold marks rates with $p < 0.05$.

Prune rate (%)	10	20	30	40	50	60	70	80	90
n	6	6	6	6	6	6	6	6	6
Gain	$+0.47$	$+0.31$	$+0.14$	$+1.02$	$\mathbf{+1.30}$	$\mathbf{+0.58}$	$+0.35$	$\mathbf{+2.33}$	$\mathbf{+3.84}$
CI low	-0.29	-0.58	-0.34	-0.02	0.64	0.12	-0.89	0.38	1.82
CI high	1.24	1.20	0.61	2.07	1.95	1.05	1.59	4.28	5.87
p	0.173	0.408	0.487	0.053	0.004	0.023	0.501	0.028	0.005
Runs positive	4/6	3/6	4/6	5/6	6/6	6/6	4/6	5/6	6/6

Section 4.3 returns to where that gain is largest.

4.2 The decomposition

Each run's final parameter count falls on a single segment of the zero-dendrite reference, so the parameter cost is exactly the local slope times the change in log parameter count (Figure 4 in Appendix B). The parameter cost is small on this architecture: at most 0.22 points through 70 percent pruning, then 1.02 at 80 percent and 1.50 at 90 percent, where the reference is steepest and dendrites add the largest share of parameters. The raw gain dominates at every rate and averages $+1.49$ points

$[0.95, 2.04]$ over the 54 runs. Dendrites are cheap here because they attach to a classifier whose input is narrow.

4.3 WHERE THE GAIN IS LARGEST

The gain is largest at the heavy prune rates, where the pruned model has lost the most training accuracy (Table 3): $+2.33$ and $+3.84$ points at 80 and 90 percent, against $+0.60$ averaged over rates 10 to 70. That is consistent with what the dendrite phase does. It fits each neuron's residual error, as in Cascade-Correlation (Fahlman & Lebiere, 1989), so a baseline that still fits its training set almost perfectly leaves a dendrite little to fit, while one that no longer fits it leaves a large residual.

5 STUDY B: A SECOND PRUNING CRITERION

Study B repeats the Study A benchmark with first-order Taylor importance (Molchanov et al., 2019) in place of group L2 magnitude, over 33 perforation runs: three seeds at each rate from 10 to 70 percent and six at 80 and 90. Every quantity of Section 3 is recorded, so Study B is measured exactly as Study A is, at matched parameter count against the zero-dendrite reference. *One protocol difference remains*: Study B's perforation runs train at learning rate 10^{-3} with a cosine schedule, where Study A uses 5×10^{-4} with plateau decay. Each run supplies its own zero-dendrite reference at its own parameter count, so the gain absorbs that difference; it shows up only in Study B's zero-dendrite reference sitting below the accuracy of the pruned checkpoints its runs start from at low prune rates, the opposite of Study A.

5.1 THE MECHANISM HOLDS UNDER A DIFFERENT CRITERION

Perforated models sit $+1.60$ points $[1.00, 2.19]$ above the zero-dendrite reference over the 33 runs, with 29 of 33 positive ($p = 9.9 \times 10^{-6}$, one-sample t-test), against $+1.15$ $[0.72, 1.58]$ in Study A. The rate mean is positive at eight of the nine prune rates, and the parameter cost stays at or below 0.83 points at every rate. Figure 2 shows the same construction as Study A's Figure 1, on identical axes, and Table 2 gives the per-rate gains; Appendix C adds the raw gain behind them.

Table 2: Study B, parameter-matched gain against the zero-dendrite reference at budget 3, by prune rate. Confidence intervals are 95 percent t-intervals; p is a two-sided one-sample t-test. Bold marks rates with $p < 0.05$.

Prune rate (%)	10	20	30	40	50	60	70	80	90
n	3	3	3	3	3	3	3	6	6
Gain	$\mathbf{+1.87}$	$\mathbf{+1.24}$	$+1.75$	$\mathbf{+1.76}$	$+0.37$	-0.16	$+1.32$	$+1.22$	$\mathbf{+3.49}$
CI low	0.74	0.52	-0.77	0.32	-1.17	-2.96	-3.06	-1.02	1.14
CI high	3.01	1.96	4.27	3.20	1.91	2.63	5.70	3.46	5.85
p	0.019	0.018	0.096	0.034	0.410	0.824	0.325	0.222	0.012
Runs positive	3/3	3/3	3/3	3/3	2/3	2/3	2/3	5/6	6/6

5.2 A STRONGER PRUNED CHECKPOINT, A SMALLER GAIN

Taylor importance produces the stronger pruned checkpoint: 51.68 percent test accuracy against 42.09 for L2 at 80 percent pruning, and 28.67 against 21.32 at 90 percent. The gains are smaller where the checkpoint is stronger: $+1.22$ at 80 percent against $+2.33$ under L2, and $+3.49$ $[1.14, 5.85]$ at 90 percent against $+3.84$, where both checkpoints are badly damaged and both gains are large. Changing the criterion changes which channels are removed and moves the pruned checkpoint by nearly ten points at 80 percent pruning, and that is consistent with Study A, where the gain is largest at the rates that cost the most accuracy.

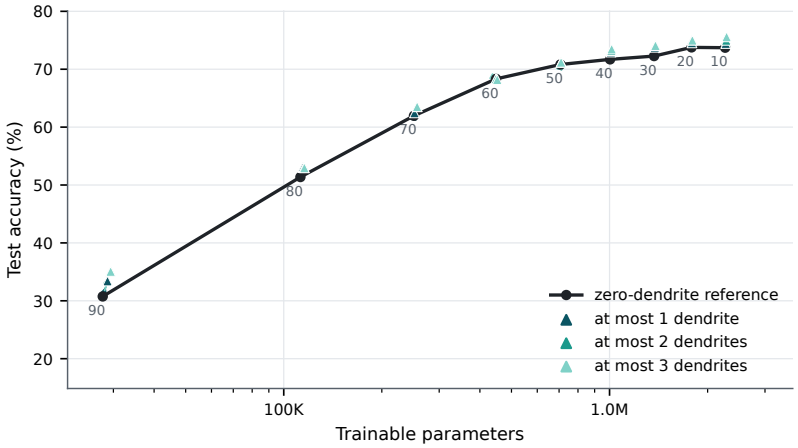

Figure 2: Study B, Taylor importance. The zero-dendrite reference (solid, circles) against trainable parameters on a log scale; labels give the prune rate. Triangles mark the mean model accuracy achieved with at most one, two or three dendrites, at its own parameter count. Axis ranges, styles and legend match Figure 1.

6 Study C: production segmentation

Study C tests the composition on a deployed model. Under successive rounds of structured pruning, perforated checkpoints sit above the zero-dendrite line at every round. These are production runs recorded without replication intervals, so we report effect sizes throughout and attach no significance tests.

Task, model and data. Study C is human detection by semantic segmentation on industrial autonomous equipment, where any pixel predicted as Person triggers a protective response downstream. Missed people and unnecessary stops both have a cost, so the primary metric is Person-class F1, computed from per-pixel precision and recall on the Person class. The model is DeepLabV3 (Chen et al., 2017) with an ASPP head: parallel atrous convolutions at several dilation rates over the backbone feature map, concatenated with image-level context from global average pooling and projected to the class count. The backbone is a substitutable component. Only the ASPP head is perforated. The images are single-channel grayscale frames of 960×600 pixels from a stereo camera mounted on the robot, lower than a typical surveillance or research capture height, recorded in warehouse, industrial, commercial and retail environments. The split has 12,606 training and 3,300 validation images, annotated as polygon masks with 10,120 Person and 18,877 Drivable Area instances. Scores are recorded at the epoch of maximum validation performance.

Pruning then perforating. We pruned the deployed model in eight successive rounds of structured pruning, saved the checkpoint after each round, and perforated it with up to one, two or three dendrites, against the same pipeline at zero dendrites (Figure 3). Every reported point is the average of five independent trials, so the comparison does not rest on a single initialization. At every checkpoint, and at every dendrite budget, the perforated models sit above the zero-dendrite model, and the remaining error falls by 1.3 to 4.3 percent. The production comparison is the one a deployment cares about: at 8.23M parameters, a third fewer than the unpruned model's 12.19M, the at-most-three-dendrite model reaches 82.65 percent Person-class F1 against 81.97 for the unpruned zero-dendrite model, and at 5.59M, 54 percent fewer, it reaches 81.89, within 0.1 points of unpruned. Each round removes between 0.5 and 2.2 million parameters while the dendrites add a few hundred, so in the terms of Equation 1 the change in $\log_{10} n$, and with it the parameter cost, is small against the spacing of the rounds.

Read across the checkpoints, the figure shows perforation helping in a different way at each stage of pruning. At the operational size, before any pruning, the dendrites already raise Person-class F1 by 0.23 points at an essentially unchanged parameter count, so the deployed model is not saturated

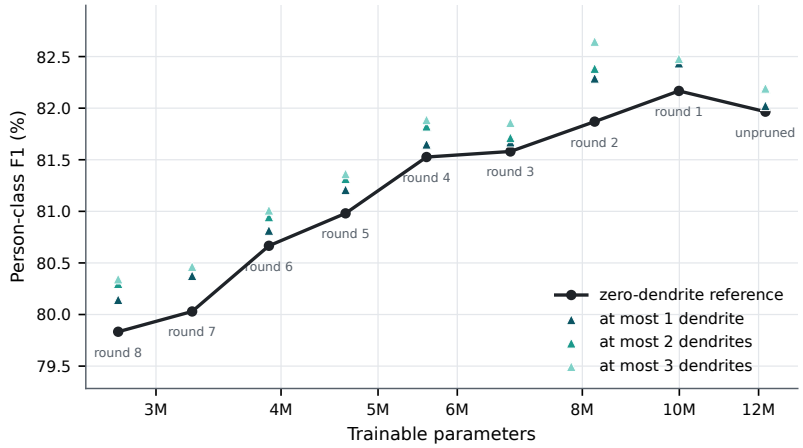

Figure 3: Study C pruning rounds, rendered from the per-checkpoint means over five independent trials. Nine checkpoints, right to left: the unpruned model and eight pruning rounds. The line is the zero-dendrite model; triangles are the at-most-N dendrite models of Section 3, drawn at their checkpoint's parameter count because the few hundred parameters a dendrite adds are invisible at this scale. The metric is Person-class F1. Axis ranges are fitted to this study; styles and legend match Figures 1 and 2.

and the added capacity still finds error it can correct. The first rounds cost the network little, and the dendrites add 0.31 points after the first round and 0.78 after the second, the largest gain across all checkpoints, which lifts the pruned model above the original deployed model at 18 and 32 percent fewer parameters. In the middle rounds, a gain of about a third of a point keeps Person-class F1 within about a tenth of a point of the unpruned level at roughly half the parameters, which is the trade a deployment is after: compression without its usual accuracy cost. In the deepest rounds, where pruning costs the most, the dendrites recover a steady 2 percent or so of the remaining error each round. Throughout, they add a few hundred parameters against the 0.5 to 2.2 million each round removes, so every point of F1 they recover comes almost free on the parameter axis, and the composition improves on pruning alone at every stage rather than only at the extremes.

7 What the three studies agree on

The three studies differ in pruning schedule, task and pruning criterion, and in all three perforation improves on the zero-dendrite baseline at matched parameter count. Under L2 pruning on Pets, the parameter-matched gain averages $+1.15$ points over 54 runs, with all nine rate means positive. Under Taylor pruning it averages $+1.60$ points over 33 runs. In the production segmentation setting, the perforated model sits above its zero-dendrite counterpart at every one of the nine checkpoints.

These gains come at little cost in size. Budget-3 models add at most 10.4 percent parameters, and under 3 percent through 70 percent pruning. Under L2 the parameter cost stays below a quarter point through 70 percent pruning, and under Taylor it never exceeds 0.83 points.

In both classification studies, perforation contributes most where pruning has taken the most. Under L2 the gain averages $+0.60$ points from 10 to 70 percent pruning, then rises to $+2.33$ at 80 percent and $+3.84$ at 90 percent. Taylor pruning leaves a checkpoint nearly ten points stronger at 80 percent, and the gain there is correspondingly smaller, $+1.22$ against $+2.33$. In the production model, perforation removes 1.3 to 4.3 percent of the remaining error at every checkpoint.

The production setting shows what this means in practice. With dendrites, the round-2 model, with 32 percent fewer parameters than the deployed network, reaches 82.65 Person-class F1 against 81.97 for the unpruned model, and at 54 percent fewer parameters it stays within 0.1 points. Across all three settings, perforation turns a pruned model into a better one at essentially the same size.

8 SCOPE

Studies A and B use half-width ResNet-18 on Oxford-IIIT Pets only, with one-shot pruning and dendrites attached to the classifier alone, the cheapest placement, and Study C perforates the ASPP head only, so the split is general while the size of each term is not, and, in the classification studies, the largest gains sit on the most heavily degraded baselines. Study C is not controlled like Study A: its pruning rounds carry no per-round intervals, and its data come from one platform and one camera configuration. Study B uses three seeds at each rate from 10 to 70 percent and six at 80 and 90, and its perforation runs use a different optimizer schedule from Study A's, which the zero-dendrite reference absorbs.

ETHICS STATEMENT

Study C trains and evaluates models on camera images of people. The images are grayscale frames recorded by a camera mounted on an industrial robot operating in warehouse, industrial, commercial and retail environments, and the models segment Person pixels so that a downstream system can respond to people near the equipment. The task is detection of people as a class; no model in this paper identifies individuals. The Study C images are not released with this paper, and this paper does not document the consent or data-handling procedures under which they were collected. Person detection feeds a safety function, and errors in either direction have consequences: a missed person is a safety risk and a false detection stops the equipment. The results here are offline accuracy comparisons and are not a validation of any model for safety-critical deployment. Studies A and B use Oxford-IIIT Pets, a public dataset of cat and dog images.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing writing, for retrieval and discovery of related work, for research ideation and execution, and for drafting sections of the paper. The experiments were designed and run by the authors, the data were produced by the authors' own training runs, and every reported number was verified by the authors against those runs or the cited source. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 defines every quantity reported, including the parameter-matched gain, the zero-dendrite reference and the decomposition. Section 4 and Appendix D give the Study A data split, model, pruning procedure, training recipe and perforation configuration; Study B changes the importance criterion and the perforation optimizer schedule. Appendix A tabulates the per-rate results behind every Study A figure, Appendix B holds the decomposition figure, Appendix C the Study B results, and Appendix E documents how dendrite counts were derived. Studies A and B use a public dataset and public libraries. Study C uses a proprietary dataset that is not released.

REFERENCES

R. Brenner and L. Itti. Perforated backpropagation: A neuroscience inspired extension to artificial neural networks. arXiv preprint arXiv:2501.18018, 2025.

R. Brenner, E. Davis, R. Chaudhari, R. Morse, J. Chen, X. Liu, Z. You, and L. Itti. Exploring the performance of perforated backpropagation through further experiments. arXiv preprint arXiv:2506.00356, 2025.

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017.

U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning (ICML)*, 2020.

S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In *Advances in Neural Information Processing Systems*, volume 2, 1989.

G. Fang, X. Ma, M. Song, M. B. Mi, and X. Wang. DepGraph: Towards any structural pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, and E. Choi. MorphNet: Fast & simple resource-constrained structure learning of deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

S. M. Jayakumar, R. Pascanu, J. W. Rae, S. Osindero, and E. Elsen. Top-KAST: Top-K always sparse training. In *Advances in Neural Information Processing Systems*, 2020.

Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 2018.

P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. Importance estimation for neural network pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam. NetAdapt: Platform-aware neural network adaptation for mobile applications. In *European Conference on Computer Vision (ECCV)*, 2018.

A PER-RATE TABLES FOR STUDY A

Table 3: The pruned checkpoints that each perforation run starts from, per prune rate. Parameters are identical across runs at each rate.

Prune rate	n	Parameters	Test acc. (%)	Train acc. (%)
0%	6	2,808,389	76.12	97.93
10%	6	2,264,295	75.93	99.20
20%	6	1,785,197	75.02	98.62
30%	6	1,370,054	74.82	98.43
40%	6	1,003,710	73.00	97.24
50%	6	706,869	71.31	96.09
60%	6	448,439	67.41	92.74
70%	6	251,005	61.08	85.08
80%	6	112,502	42.09	59.98
90%	6	27,822	21.32	24.48

Table 4: Study A, decomposition against the zero-dendrite reference at budget 3, per prune rate, in points. The gain equals raw gain minus parameter cost (Equation 1).

Prune rate	n	Gain vs. Z	Raw gain	Parameter cost	Params added
10%	6	+0.47	+0.47	0.000	0.8%
20%	6	+0.31	+0.31	0.002	1.1%
30%	6	+0.14	+0.17	0.034	1.3%
40%	6	+1.02	+1.09	0.065	1.3%
50%	6	+1.30	+1.39	0.089	1.8%
60%	6	+0.58	+0.74	0.158	2.5%
70%	6	+0.35	+0.57	0.222	2.1%
80%	6	+2.33	+3.36	1.024	4.8%
90%	6	+3.84	+5.34	1.498	10.4%

B Additional figure for Study A

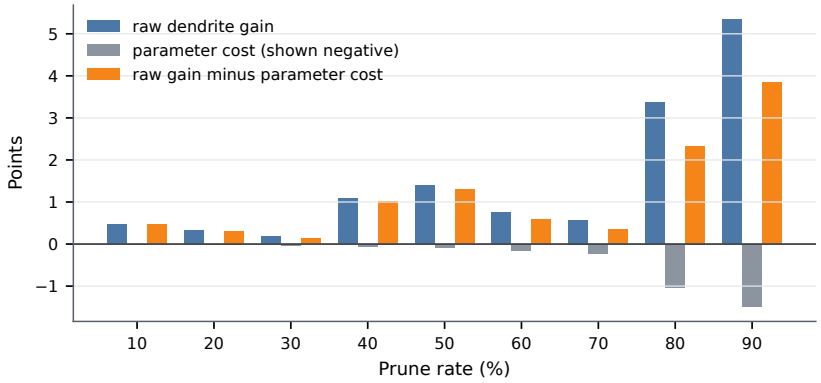

Figure 4: Study A, per prune rate: the raw dendrite gain g, the parameter cost c (drawn negative), and the difference $g - c$, which is the parameter-matched gain of Equation 1. The parameter cost is below a quarter of a point through 70 percent pruning.

C Per-rate results for Study B

Table 5: Study B, per prune rate, measured as in Study A: parameter-matched gain against the zero-dendrite reference at budget 3. Accuracies in percent, gains in points. Intervals are 95 percent t-intervals and p is a two-sided one-sample t-test.

Rate	Seeds	Pruned checkpoint	Zero-dendrite	Gain	95% CI	p	Raw gain
10%	3	75.62	73.72	+1.87	[+0.74, +3.01]	0.019	+1.87
20%	3	74.62	73.77	+1.24	[+0.52, +1.96]	0.018	+1.24
30%	3	74.04	72.27	+1.75	[−0.77, +4.27]	0.096	+1.81
40%	3	73.08	71.70	+1.76	[+0.32, +3.20]	0.034	+1.78
50%	3	72.05	70.82	+0.37	[−1.17, +1.91]	0.410	+0.38
60%	3	68.74	68.33	−0.16	[−2.96, +2.63]	0.824	−0.12
70%	3	63.17	61.96	+1.32	[−3.06, +5.70]	0.325	+1.56
80%	6	51.68	51.38	+1.22	[−1.02, +3.46]	0.222	+1.59
90%	6	28.67	30.75	+3.49	[+1.14, +5.85]	0.012	+4.32

D PROTOCOL DETAILS

Training. The unpruned width-0.5 model is trained with AdamW at learning rate 10^{-3}, and pruned models are fine-tuned with AdamW at 5×10^{-4}. Both use weight decay 10^{-4}, batch size 32, no label smoothing, `ReduceLROnPlateau` on validation accuracy (mode max, factor 0.1, patience 3, minimum learning rate 0.01 times the initial rate), and early stopping with patience 8 and minimum improvement 0.001. No run has an epoch cap. Test accuracy is reported at the best-validation epoch with weights restored to that epoch. Training transforms are `RandomResizedCrop(224, scale=(0.5, 1.0))` and random horizontal flip; evaluation uses `Resize(256)` and `CenterCrop(224)`; both normalize with ImageNet statistics.

Pruning. Torch-Pruning `MagnitudePruner` over DepGraph groups with `GroupMagnitudeImportance(p=2)`, `global_pruning=False`, the classifier in `ignored_layers`, applied once at the target rate.

Perforation. `perforated_backpropagation=True`, `max_dendrites=3`, `max_dendrite_tries=3`, `retain_all_dendrites=False`, `module_names_to_perforate=["Linear"]`, `output_dimensions=[-1, 0]`. The perforation run starts from the converged pruned checkpoint and uses the fine-tuning optimizer and schedule, re-initialized at the start and after every restructure. Perforation runs last 155 to 371 epochs in total (median 246). A separate arm with one attempt per dendrite was run and is excluded from every result. All Study A perforation runs use `perforatedai` 3.2.3 with the `perforatedbp` backend 3.2.3, and results may differ under other versions.

E DENDRITE-COUNT BOOKKEEPING

Dendrite counts are derived from parameter counts rather than read from the library's integrated-dendrite counter, which disagrees with the parameter change in 30 of the 54 runs. Each dendrite adds one copy of the classifier's weights and bias, `in_features` $\times\, 37 + 37$ parameters: 8,547, 7,585, 6,660, 5,698, 4,773, 3,811, 2,849, 1,924 and 962 at 10 through 90 percent pruning, confirmed from consecutive architectures in every run. The derived count is $(n_{\text{final}} - n_{\text{start}})$ divided by that step, an integer in every run. Of the 54 runs, 40 kept three dendrites, 6 kept two, 6 kept one and 2 kept none.