Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Hidden Classifier Collapse in Medical Image Classification: The Impact of Structured Pruning, Data Scarcity, and Post-Pruning Optimization

Zulqarnain Ali
Yeungnam University
Isabel de la Torre Dıez
Universidad Europea del Atlántico
Park Changhyun
Yeungnam University
Anny Marıa Hernandez Cuello
Universidad Europea del Atlántico
Yanelis Manzano Nieves
Universidade Internacional do Cuanza
M. Zeeshan Jhandir
Yeungnam University
Hafiz Muhammad Raza ur Rehman
mrazaurrehman@ynu.ac.kr

Yeungnam University

Article

Keywords: structured pruning, medical image classification, data scarcity, classifier collapse, batch normalization, model compression, optimization

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11705806/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Hidden Classifier Collapse in Medical Image Classification: The Impact of Structured Pruning, Data Scarcity, and Post-Pruning Optimization

Zulqarnain Ali[1], Isabel de la Torre Díez[2], Park Changhyun[1], Anny María Hernández Cuello[2,3,4,5], Yanelis Manzano Nieves[5,6,7], M. Zeeshan Jhandir[1*], Hafiz Muhammad Raza ur Rehman[1*]

[1]Yeungnam University, Gyeongsan, Republic of Korea .
[2]Universidad Europea del Atlántico, Isabel Torres 21, 39011, Santander, Spain .
[3]Universidad Internacional Iberoamericana, Campeche, 24560, Mexico .
[4]Universidad Internacional Iberoamericana, Arecibo, 00613, Puerto Rico, USA .
[5]Universidade Internacional do Cuanza, Cuito, Bié, Angola .
[6]Fundación Universitaria Internacional de Colombia, Bogotá, Colombia .
[7]Universidad de La Romana, La Romana, Dominican Republic .

*Corresponding author(s). E-mail(s): zeeshanjhandir@live.com; mrazaurrehman@ynu.ac.kr;
Contributing authors: zulqar445ali@gmail.com; isator@uva.es; park@yu.ac.kr; anny.hernandez@unic.co.ao; Yanelis.manzano@unic.co.ao;

Abstract

Structured pruning reduces the computational complexity of medical image classifiers but may compromise classification reliability under limited training data, with aggregate accuracy or AUC potentially concealing severe prediction failures. This study investigates hidden classifier collapse following one-shot structured pruning under controlled data scarcity using five MedMNIST v2 datasets, ResNet-18 and ResNet-34 architectures (with ResNet-50 in additional controls), four sparsity levels (70-95%), and repeated random seeds. Fixed and adaptive post-pruning protocols were compared, jointly varying batch size, learning rate, and optimization update budget, while focused controls examined update count,

learning-rate scaling, BatchNorm running-statistic adaptation, optimizer family, and pruning criterion. The protocols showed substantial differences on PneumoniaMNIST and BloodMNIST, sparsity-dependent differences on OrganSMNIST and RetinaMNIST, and no detectable difference on DermaMNIST. Additional optimization updates, learning-rate scaling, and BatchNorm-statistic adaptation each reduced classifier collapse in selected conditions, although no single factor was identified as universally necessary. Conventional accuracy and AUC could remain apparently favorable despite hard-label collapse, whereas balanced accuracy and Matthews correlation coefficient exposed classification failures. The adaptive optimization strategy also showed inconsistent performance across optimizer families. These findings emphasize the importance of class-sensitive evaluation, collapse indicators, and appropriate post-pruning adaptation for assessing the reliability of compressed medical image classifiers.

Keywords: structured pruning; medical image classification; data scarcity; classifier collapse; batch normalization; model compression; optimization

1 Introduction

Deep convolutional networks are widely used for medical image classification, but their computational cost often exceeds what point-of-care and edge deployments can afford. Structured pruning, which removes entire filters or channels, reduces this cost with real speed-ups on standard hardware; unstructured pruning typically needs specialized sparse kernels to realize any speed-up [1–4]. Medical imaging differs from the natural-image regimes in which these techniques were developed: labeled data is expensive, tasks are often class-imbalanced, and the number of independent patients, not image patches, governs generalization [5, 6]. Prior studies report accuracy degradation when aggressive pruning meets scarce training data [6] and disproportionate harm to tail classes [7]. A pruned model that has degraded to a near-constant predictor can still look acceptable on aggregate metrics, which could be a safety risk if such a model were deployed clinically without further validation.

The closest prior joint study of pruning ratio and training-set size is Götz et al. [6], conducted on histopathology cell images, where several training images may come from the same patient. That setting differs from whole-image, one-sample-per-patient datasets. STAMP [5] prunes and trains simultaneously for brain and cardiac MRI segmentation, a different pruning regime (train-time, not post-hoc) from the one-shot pruning studied here. Holste et al. [7] and Larsson & Odin [8] each vary a single axis (class imbalance or sparsity) rather than the joint interaction. None of these works reports whether the optimizer configuration (batch size, learning rate, or BatchNorm behavior [9]) was adapted to the very small sample counts produced by extreme scarcity. Below we place the study within the recent literature and the pruning, data-scarcity and large-batch-training work it builds on, and then state the research questions that follow from this gap.

1.1 Recent evidence base

Recent work from 2022-2025 bears on the present study from several directions: efficient inference, structured and unstructured pruning, model compression, medical-image classification and segmentation, calibration, class imbalance, and resource-constrained deployment. It includes pruning methods, compression surveys, efficient vision models, biomedical benchmarks, medical-image learning under limited or imbalanced data, and work on evaluation reliability[10–32]. We draw on these papers to place the contribution narrowly. The study does not propose a new pruning algorithm; it examines whether the evaluation protocol and metric choice can conceal hard-label collapse after one-shot structured pruning in scarce-data medical classification.

1.2 Structured pruning

Han et al. [1] introduced the train-prune-retrain paradigm. Structured pruning subsequently focused on removing complete filters or channels: L1-norm filter pruning [3] ranks and removes filters by weight magnitude in a single pass (one-shot), while Network Slimming [4] uses batch-norm scaling factors and can be applied iteratively across multiple prune-retrain cycles. One-shot pruning removes the full target sparsity at once and fine-tunes once; iterative pruning removes sparsity in smaller increments across repeated retraining cycles and often better preserves accuracy at very high sparsity, at higher compute cost. All pruning in this paper is one-shot; our conclusions do not extend to iterative schedules (see 'Limitations' in the Discussion). Gale et al. [33] found simple magnitude criteria competitive with more complex methods, and Liu et al. [34] found that randomly re-initialized pruned architectures can match trained-then-pruned ones on standard benchmarks. That finding concerns training from scratch after pruning, which differs from the random *filter-selection* control we use (Results, 'Pruning criterion'), which prunes by random rather than magnitude criterion but still fine-tunes from the same pretrained-then-pruned weights; we cite [34] as the motivation for including a random-selection control, not as an identical experimental design. Blalock et al. [35] document inconsistent evaluation protocols across the pruning literature. DepGraph [36] automates dependency-aware structured pruning for arbitrary architectures and is the pruning engine used here.

1.3 Compensation, BatchNorm re-estimation and data-free recovery

Fine-tuning is not the only way to recover a one-shot pruned network. Closed-form compensation methods model the removed channels as affine functions of the retained ones and fold the correction into the remaining weights using a small unlabelled calibration set; CORP argues that the error of the removed channels, rather than a poor importance ranking, drives one-shot collapse, although it is evaluated on vision transformers, not ResNets [37]. Re-estimating BatchNorm statistics without gradient steps adapts a network to a new data distribution [38], and the same recalibration is used to evaluate pruned candidate sub-networks fairly [39]; our BatchNorm decomposition tests the same mechanism after pruning. Data-free recovery goes one step further by

synthesizing inputs that match the stored BatchNorm statistics and using them for distillation or pruning [40]. We do not compare against these methods; our contribution is to show how the fine-tuning protocol and the choice of metric can hide collapse.

1.4 Data scarcity in medical pruning

Prior work conflates several distinct notions of "scarcity": low patient count, low total image count, class imbalance, and patch-based sampling that inflates apparent sample counts relative to patient counts. These are not interchangeable. STAMP [5] studies segmentation from very few labelled images with a simultaneous train-and-prune schedule. Götz et al. [6] jointly sweep parameter count and training-set size on histopathology cell images, and report accuracy degradation at extreme scarcity attributed to the joint sparsity-scarcity interaction, not explicitly to optimizer configuration. Holste et al. [7] study class imbalance (long-tailed distributions) under pruning, a different axis from total sample count. Larsson & Odin [8] sweep sparsity at fixed sample count. We use whole, one-sample-per-patient images at controlled fractions of the training set, isolating total sample count from the class-imbalance and repeated-patient confounds that may be present in prior work (Table 1). We use MedMNIST v2 [41] as the benchmark.

1.5 Large-batch training and learning-rate scaling

A separate literature addresses making *large*-batch training work as well as small-batch training. Goyal et al. [42] propose linear learning-rate scaling with batch size, validated for SGD on large-batch ImageNet training. LARS [43] and LAMB [44] extend this with per-layer adaptive scaling to stabilize training at batch sizes where a single global learning rate becomes unstable. For Adam-type optimizers, theory instead suggests a square-root scaling rule [45], and recent work on small-batch language-model training keeps the half-life of Adam's second-moment average fixed when the batch size changes [46]. This paper studies the inverse regime: *small*, data-scarcity-forced batches during post-pruning fine-tuning. Our adaptive protocol follows none of these rules. It *raises* the fine-tuning learning rate as the batch shrinks ($LR_{ft} = LR_{base} \cdot 128/\text{eff_batch}$, a factor of 11.6 at a batch of 11), which is the opposite direction to linear scaling and to the square-root rule. It is an empirical heuristic, and we make no claim of theoretical justification. The study asks whether the unscaled base learning rate is itself a failure mode (collapse) when only a handful of optimizer updates per epoch are available, not only a convergence-speed concern. The Results subsection 'Step-count, BatchNorm and optimizer controls' reports empirical results under SGD, LARS, and LAMB as a direct test of whether the collapse-and-recovery pattern is specific to AdamW.

1.6 Research questions and contributions

Because prior work does not report whether post-pruning optimizer settings were adapted to extreme scarcity, we treat them here as an experimental variable alongside sparsity and training-set size. Four research questions guide the study:

Table 1 Comparison with representative prior work.

Study	Datasets	Scarcity	Opt. ctrl.	#DS
Götz et al. [6]	Histopathology	Training-set size	No	1
STAMP [5]	Brain, cardiac MRI	Few labelled images	No	1
Holste et al. [7]	Chest X-ray	Class imbalance	No	1
Larsson & Odin [8]	HAM10000	Fixed n	No	1
This work	MedMNIST v2	Whole-image, patient-level	Yes	5

1. **RQ1:** Does structured pruning produce hard-label classifier collapse under severe whole-image medical data scarcity, and how does the rate vary across datasets and sparsity levels?
2. **RQ2:** Can aggregate metrics such as AUC and Accuracy detect collapse, or are class-sensitive metrics required?
3. **RQ3:** Under the tested protocol, is collapse associated more strongly with nominal sparsity or with post-pruning adaptation choices involving batch size, learning rate, update count, and BatchNorm statistics?
4. **RQ4:** Do the observed patterns remain similar across ResNet depths, optimizer families, and pruning criteria?

The study makes four contributions. First, it provides a repeated-seed evaluation across five MedMNIST v2 datasets and up to three ResNet depths, comparing fixed and data-scaled (adaptive) post-pruning protocols at each dataset's most extreme tested scarcity level. Second, it shows that the fixed protocol is associated with collapse on three of five datasets, while the adaptive protocol reduces collapse in the tested cells; because several training variables change together, this is a protocol-level result. Third, focused controls examine update budget, learning-rate scaling, BatchNorm statistics, optimizer family, and pruning criterion, and we do not claim that any single factor is universally necessary. Fourth, AUC and Accuracy can remain apparently stable during hard-label collapse, so we recommend reporting Balanced Accuracy, MCC, per-class recall, and the number of unique predicted labels alongside aggregate metrics. Negative and mixed findings, including the absence of a consistent cross-architecture advantage and the lack of a uniform L1-versus-random advantage, are reported rather than omitted.

The Results section follows the order of the questions: collapse across datasets and sparsity levels (RQ1), the masking of collapse by aggregate metrics (RQ2), the factors associated with collapse (RQ3), and robustness across architectures, pruning criteria and optimizers (RQ4). Fig. 1 summarizes the pipeline, and the Methods section gives the full experimental details.

2 Methods

2.1 Datasets

We used five 2-D MedMNIST v2 datasets [41] at 28×28 resolution, with one image per patient and no patch mining (Table 2). Only the training split is subsampled to create scarcity conditions; validation and test splits are always used at full size, unchanged across all fractions, pipelines, sparsity levels, and seeds. A given seed's stratified training subsample is the same set of images across all sparsity levels and all pipeline conditions for that (dataset, fraction, architecture, seed) combination, so pruning and optimization conditions are compared on identical training data; a new baseline model is trained independently per (dataset, fraction, architecture, seed) combination, and each control condition (fixed, adaptive, step-count, BatchNorm variants, optimizer variants) fine-tunes from the same pruned checkpoint derived from that one baseline, so conditions within a (dataset, fraction, architecture, seed) cell share an identical pruned starting point. A per-dataset floor of five expected samples in the rarest class is enforced when choosing the lowest training fraction: this is why DermaMNIST and RetinaMNIST are tested down to 10% rather than 1% (their rarest classes are proportionally small enough that 1% would leave under one expected sample of that class).

Table 2 Dataset summary. K denotes the number of classes and $N_{\min}$ the training-image count at the lowest tested fraction.

Dataset	K	Train	Lowest	$N_{\min}$	Ratio
PneumoniaMNIST	2	4,708	1%	47	1:2.9
DermaMNIST	7	7,007	10%	700	1:58.7
BloodMNIST	8	11,959	1%	119	1:2.7
OrganSMNIST	11	13,932	1%	139	1:5.6
RetinaMNIST	5	1,080	10%	108	1:7.4

Imbalance ratio is the count of the most common class divided by the count of the rarest class in the full training split.

2.2 Architectures and training protocols

We used ResNet-18/34/50 [48] from torchvision, adapted for 28×28 input by replacing the 7×7 stride-2 stem with a 3×3 stride-1 convolution and removing the initial max-pool (Fig. 1). The main factorial grid (Results, 'Fixed-protocol collapse by sparsity level' to 'Interaction regression') uses ResNet-18 and ResNet-34; ResNet-50 is used for the cross-architecture comparison (Results, 'Cross-architecture comparison') and the step-count, BatchNorm and optimizer controls (Results, 'Step-count, BatchNorm and optimizer controls'), but not the main grid, for compute reasons.

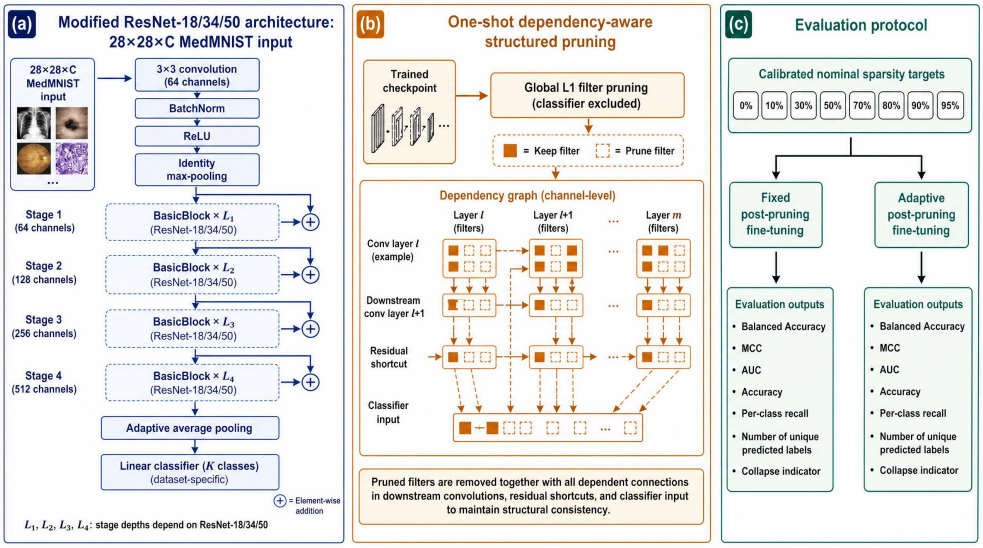

Fig. 1 Architecture and evaluation protocol for modified ResNet models, dependency-aware one-shot L1 pruning, and post-pruning evaluation.

We compared training configurations that differ in batch size, learning rate, step count, and (in one control) BatchNorm behavior, so they differ in more than one respect. **Fixed**: batch size capped at 128 (PyTorch's default `DataLoader` behavior is used, i.e. `drop_last=False`, so the final batch in an epoch may contain fewer than 128 samples when the training subset is not an exact multiple; at PneumoniaMNIST's 1% fraction, $N_{\min} = 47$, this yields exactly one batch of 47 samples per epoch, not "fewer than one batch"). **Adaptive**: batch size set to eff_batch $= \min(128, \max(8, \lfloor n_{\text{train}}/4 \rfloor))$, chosen to target approximately four optimizer updates per epoch across the range of training-set sizes we test, with the fine-tuning learning rate raised to $LR_{ft} = LR_{base} \cdot 128/\text{eff_batch}$ ($LR_{base} = 10^{-3}$). All protocols fine-tune for 15 epochs with AdamW (weight decay 5×10^{-4}) and cosine annealing, so a cell has $\lceil n_{\text{train}}/\text{batch} \rceil$ updates per epoch and $15\lceil n_{\text{train}}/\text{batch} \rceil$ updates in total (Table 3). **Step-count-only** (Results, 'Step-count, BatchNorm and optimizer controls'): batch size and learning rate held at their fixed-condition values, but fine-tuning epoch count raised so that the total optimizer-step count matches the adaptive condition's step count for that cell (derived per cell from n_{train} and eff_batch, not a single constant across datasets; for PneumoniaMNIST at 1% data this is 75 epochs at batch 128, matching adaptive's $5 \times 15 = 75$ steps). The CosineAnnealingLR scheduler is stepped once per epoch in all conditions, so raising epoch count under step-count-only also lengthens the LR schedule in epoch units; we did not additionally re-derive a step-matched schedule shape, which is a scope limitation noted in 'Limitations' in the Discussion. We did not directly test eff_batch = 11 specifically (PneumoniaMNIST's exact heuristic value at 1% data) in the 5-seed sweep; the batch-size sweep in Results, 'Batch-size sensitivity' brackets it with tested values of 8 and 16, and a later 15-seed repeat includes it.

Table 3 Fine-tuning settings implied by the fixed and adaptive protocols at each dataset's lowest tested fraction (ResNet-18 baseline, $LR_{base} = 10^{-3}$, 15 epochs). Where eff_batch equals 128 the two protocols coincide.

Dataset	n_{train}	eff_batch	LR_{ft}	Updates, fixed	Updates, adaptive
Pneumonia (1%)	47	11	1.16×10^{-2}	15	75
Derma (10%)	700	128	1.00×10^{-3}	90	90
Blood (1%)	119	29	4.41×10^{-3}	15	75
Organs (1%)	139	34	3.76×10^{-3}	30	75
Retina (10%)	108	27	4.74×10^{-3}	15	60

For DermaMNIST at 10% data the heuristic returns eff_batch = 128, so the fixed and adaptive protocols are identical there. The differences we observe between the two Derma runs (a pooled Balanced Accuracy difference of +0.002, and up to ±0.03 at individual sparsity levels) therefore estimate run-to-run variation at the 5-seed level.

2.3 Pruning and calibration

Torch-Pruning with DepGraph [36] performs dependency-grouped one-shot structural pruning of the trained baseline. L1 filter norm [3] (rank filters by L1 weight magnitude, remove the smallest) is the primary criterion; random filter selection is a control, both applied via the same DepGraph dependency-grouping machinery so the same groups of coupled channels are removed together under either criterion. Nominal sparsity targets {0,10,30,50,70,80,90,95}% are calibrated against measured post-prune parameter-count reduction via interpolation, per architecture (Table 4 reports the ResNet-18 case; we did not recompute this table for ResNet-34/50, so absolute achieved sparsity for those architectures may differ slightly from the ResNet-18 values shown, though the calibration procedure is identical). "Sparsity" throughout this paper refers to fraction of parameters removed, not fraction of channels removed (which differs due to dependency grouping) or FLOPs reduction, which Table 4 reports separately as multiply-accumulate operations (MACs). Conclusions are limited to one-shot L1/random pruning; we did not test iterative pruning schedules.

Table 4 Sparsity calibration for ResNet-18 on PneumoniaMNIST. Achieved ρ is the mean measured parameter reduction in the experiments. MACs for a 28×28 input were re-measured on five trained baselines pruned to the same parameter sparsity (mean; SD across baselines about 5, 3 and 2 M at 50, 90 and 95%).

Nominal	Achieved ρ (params)	Params (M)	MACs (M)
0%	0.000	11.17	457.2
50%	0.505	5.53	202.2
90%	0.902	1.10	59.3
95%	0.951	0.55	37.6

At 95% sparsity the pruned network keeps about 8% of the multiply-accumulate operations of the dense model (37.6 of 457.2 M), so parameter sparsity translates into a smaller reduction in arithmetic than in size; we did not measure latency.

2.4 Metrics and collapse definitions

We report Balanced Accuracy (primary), MCC, Sensitivity, Specificity, AUC-ROC, per-class recall, and the number of unique predicted labels on the held-out test set. For multiclass tasks Balanced Accuracy is the mean per-class recall, Sensitivity is macro-averaged recall, Specificity is the macro-average of one-vs-rest specificities, F1 is macro-averaged, MCC is the multiclass generalization of the Matthews correlation coefficient (as implemented in scikit-learn), and AUC-ROC is the one-vs-rest macro average (for the two-class dataset, the area for the positive-class probability). We use Balanced Accuracy and MCC because, unlike ROC-AUC, they depend on the final predicted labels; precision-recall based summaries are also known to be more informative than ROC curves under class imbalance [49], and MCC is preferred over accuracy and F1 for imbalanced binary classification [50]. Macro precision-recall AUC was logged only in some control runs and is not analysed. **Total collapse**: the model predicts only one class for every test sample ($n_{\text{unique_preds}} = 1$); this criterion does not separately flag a model that predicts the majority class on, say, 95% of samples without reaching exactly 100%. We do not have a systematic entropy or majority-class-percentage statistic computed across the whole grid, and note this as a scope limitation (see 'Limitations' in the Discussion); where we discuss "near-collapse" informally in Results, we mean $n_{\text{unique_preds}} > 1$ but Balanced Accuracy near the chance floor. **Partial collapse (class abandonment)**: a class is considered abandoned when its per-class recall on held-out test data is below 0.05. We call this criterion *pre-specified* rather than *pre-registered*, since no public registration preceded the experiments. This criterion is applied only to DermaMNIST in this paper (Results, 'Minority-class abandonment in DermaMNIST'); it has not been applied to the other four datasets.

2.5 Statistical analysis

Our primary reporting is per-sparsity-level (Results, 'Fixed-protocol collapse by sparsity level'), not pooled across sparsity levels, because pooling four sparsity levels $\times$ five seeds into one $n = 20$ sample treats repeated evaluations of the same baseline checkpoint and the same seed's training data as independent observations, which they are not. Where we report a pooled Fisher's exact test across the four highest sparsity levels (Table 6, following the per-sparsity detail in Table 5), we label it explicitly as a descriptive pooling, not an independent-sample test, and the per-sparsity tables are the primary evidence. We did not fit a mixed-effects or repeated-measures model with seed as a random effect; this is noted under 'Limitations' in the Discussion as the more statistically appropriate analysis we have not yet performed. Balanced-Accuracy differences are reported with 5,000-resample bootstrap 95% CIs. An OLS interaction regression, Balanced Accuracy $\sim$ sparsity $\times$ fraction, is fit per dataset for descriptive purposes; it does not survive Benjamini-Hochberg correction across datasets and is reported briefly in Results ('Interaction regression') rather than treated as a supported

finding. We do not report a DeLong test: it requires paired per-sample predictions, which our per-run logs do not retain.

3 Results

We first compare the fixed and adaptive protocols across datasets and sparsity levels, then look at metric behavior during collapse, and finally turn to the controls that probe which post-pruning choices matter. Unless stated otherwise, "collapse" means that the model predicts a single class for every test image (Methods, 'Metrics and collapse definitions'), "Balanced Accuracy" (BA) is the primary metric, and sparsity is the nominal fraction of parameters removed (Methods, 'Pruning and calibration').

3.1 Fixed-protocol collapse by sparsity level

This subsection addresses RQ1 and the first part of RQ3: how often does the fixed protocol collapse, and does the adaptive protocol prevent it? Table 5 reports collapse count and mean BA separately at each of the four highest tested sparsity levels, at each dataset's lowest tested fraction, ResNet-18, fixed configuration. Collapse is not uniform across sparsity within a dataset: PneumoniaMNIST collapses at all four levels (5/5 throughout); BloodMNIST and RetinaMNIST show partial collapse that varies by level (3-4/5 for Blood, 3-4/5 for Retina) rather than a clean threshold; OrganSMNIST shows collapse only at 80% (1/5) and not at the other three levels; DermaMNIST shows no collapse at any level tested under the fixed configuration.

Table 5 Fixed protocol: collapse counts and Balanced Accuracy by sparsity.

Dataset (frac, chance)	70%	80%	90%	95%
Pneumonia (1%, 0.500)	5/5, 0.500	5/5, 0.500	5/5, 0.500	5/5, 0.500
Derma (10%, 0.143)	0/5, 0.277	0/5, 0.293	0/5, 0.266	0/5, 0.268
Blood (1%, 0.125)	4/5, 0.148	3/5, 0.148	4/5, 0.138	4/5, 0.142
Organs (1%, 0.091)	0/5, 0.363	1/5, 0.296	0/5, 0.278	0/5, 0.236
Retina (10%, 0.200)	3/5, 0.200	3/5, 0.233	4/5, 0.208	4/5, 0.204

Pooling across these four levels descriptively (Table 6), with the repeated-measures caveat described in Methods ('Statistical analysis'), PneumoniaMNIST, BloodMNIST, and RetinaMNIST show substantial collapse under the fixed configuration, whereas DermaMNIST and OrganSMNIST show little to none. The adaptive configuration (Table 7) shows zero collapse at every sparsity level on Pneumonia, Blood, Derma, and Organs, with Retina reduced but not eliminated (0-2/5 across levels).

Fisher's exact test on the pooled collapse counts (fixed vs. adaptive), Benjamini-Hochberg corrected across 5 datasets: PneumoniaMNIST $p = 7.3 \times 10^{-11}$, BloodMNIST $p = 1.9 \times 10^{-6}$, RetinaMNIST $p = 0.017$ (significant); DermaMNIST, OrganSMNIST $p = 1.0$. Because the pooled observations are not independent (Methods, 'Statistical analysis'), we report these tests as descriptive rather than as the

Table 6 Descriptive pooling across the four highest sparsity levels.

Dataset (frac)	Fixed coll.	Fixed BA	Adapt. coll.	Adapt. BA
Pneumonia (1%)	20/20	0.500	0/20	0.687
Derma (10%)	0/20	0.276	0/20	0.278
Blood (1%)	15/20	0.144	0/20	0.635
Organs (1%)	1/20	0.293	0/20	0.431
Retina (10%)	14/20	0.211	5/20	0.279

Table 7 Adaptive protocol: collapse counts and Balanced Accuracy by sparsity.

Dataset (frac)	70%	80%	90%	95%
Pneumonia (1%)	0/5, 0.705	0/5, 0.715	0/5, 0.687	0/5, 0.641
Derma (10%)	0/5, 0.302	0/5, 0.285	0/5, 0.291	0/5, 0.235
Blood (1%)	0/5, 0.636	0/5, 0.652	0/5, 0.633	0/5, 0.619
Organs (1%)	0/5, 0.441	0/5, 0.435	0/5, 0.436	0/5, 0.411
Retina (10%)	2/5, 0.265	2/5, 0.259	1/5, 0.287	0/5, 0.305

primary statistical evidence, which is the per-sparsity-level counts in Tables 5-7. Bootstrap 95% CIs on the pooled Balanced-Accuracy difference (Table 8) exclude zero for 4 of 5 datasets. Fig. 2 summarizes the same contrast. Collapse under the fixed protocol is therefore dataset-dependent and largely absent under the adaptive protocol. The next subsection asks whether aggregate metrics reveal it.

Table 8 Bootstrap 95% confidence intervals for adaptive-minus-fixed Balanced Accuracy.

Dataset	Mean diff.	95% CI
Pneumonia	+0.187	[+0.151, +0.225]
Derma	+0.002	[−0.019, +0.023]
Blood	+0.491	[+0.467, +0.512]
Organs	+0.137	[+0.096, +0.181]
Retina	+0.068	[+0.038, +0.097]

3.2 Metric masking

This subsection addresses RQ2: can aggregate metrics detect the collapse described above? Using PneumoniaMNIST at 90% sparsity, 1% data and the fixed configuration (Fig. 3), AUC-ROC and Accuracy remain visibly higher than Balanced Accuracy and MCC across the sparsity sweep, with Balanced Accuracy and MCC reaching the chance floor (0.500, 0.000) while AUC and Accuracy still appear informative.

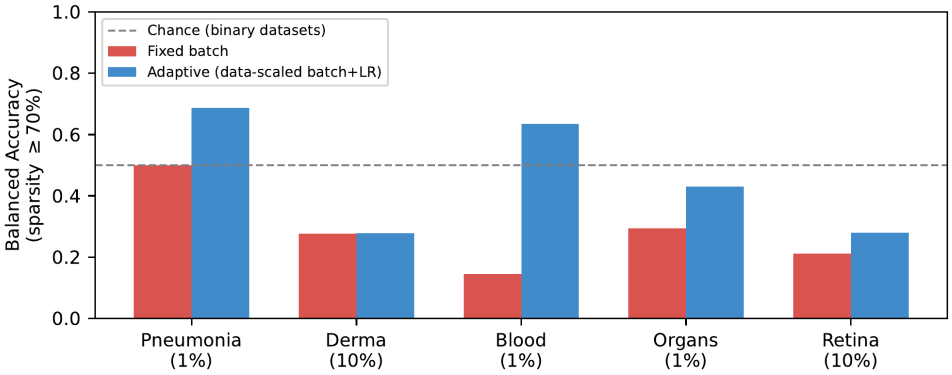

Fig. 2 Mean Balanced Accuracy at high sparsity under fixed and adaptive protocols.

This is not because AUC is computed incorrectly: AUC measures whether predicted *scores* rank positive cases above negative cases across all thresholds, and a model can preserve a useful score ranking while its arg-max decision rule collapses to a single class. AUC reflects ranking quality, not final label correctness. Accuracy on this class-imbalanced binary task (the majority class is 62.5% of the test split) is inflated toward the majority-class baseline once the model always predicts that class. Balanced Accuracy and MCC use the final predicted label directly and are not subject to either effect, which is why they reveal what AUC and Accuracy do not. Class-sensitive metrics are therefore required to detect collapse. The following subsections turn to its possible causes.

3.3 Learning-rate scaling at matched step count

The first control for RQ3 asks whether learning-rate scaling alone, without changing batch size or step count, prevents collapse. At each dataset's most extreme cell (95% sparsity), a learning-rate-only (LR-only) control (identical batch size and gradient-step count to the fixed configuration, with only the learning rate scaled) leaves 0/5 degenerate (collapsed) seeds on Pneumonia, Organs and Derma, 1/5 on Blood and 2/5 on Retina (Table 9), versus 5/5, 4/5, 0/5, 4/5, and 0/5 degenerate under the matched fixed condition on Pneumonia, Blood, Organs, Retina, and Derma respectively. On DermaMNIST and OrganSMNIST the fixed condition does not collapse at this cell, so there is no collapse for the LR-only control to recover from; the control shows non-degeneracy on those two datasets, not recovery. Of the three datasets that do collapse under the fixed condition, learning-rate scaling removes collapse completely on Pneumonia and reduces it on Blood (4/5 to 1/5) and Retina (4/5 to 2/5). Nor is a low degenerate count equivalent to good performance: BloodMNIST's LR-only mean Balanced Accuracy is 0.232, well below chance-adjacent usefulness for an 8-class task, and one of its seeds sits exactly at the chance level of 0.125.

This result alone does not show that learning-rate mis-specification, rather than step count, is the dominant mechanism. A step-count-only control (Results, 'Step-count, BatchNorm and optimizer controls') showed that increasing steps alone, at the

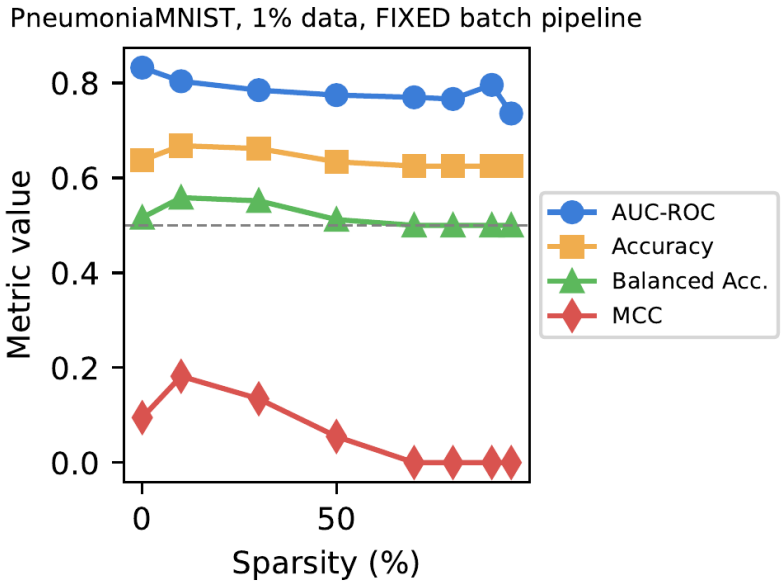

Fig. 3 Aggregate metrics during hard-label collapse on PneumoniaMNIST.

Table 9 Learning-rate-only control versus the matched fixed protocol.

Dataset (frac)	Fixed degen.	LR-only degen.	LR-only mean BA
Pneumonia (1%)	5/5	0/5	0.619
Blood (1%)	4/5	1/5	0.232
Organs (1%)	0/5	0/5	0.365
Retina (10%)	4/5	2/5	0.239
Derma (10%)	0/5	0/5	0.255

fixed unscaled learning rate, also prevents collapse, so learning-rate scaling is not the sole factor; see that subsection and the Discussion.

3.4 Interaction regression

To test whether the effect of sparsity depends on the training fraction (RQ3), we fitted Balanced Accuracy $\sim$ sparsity $\times$ fraction per dataset in the adaptive configuration. Only PneumoniaMNIST is nominally significant ($\beta = +0.077$, $p = 0.027$), and this does not survive Benjamini-Hochberg correction across the 5 datasets ($p_{BH} = 0.134$). We do not have statistically defensible evidence of this interaction in any dataset after correction; full per-dataset coefficients are in the supplementary material accompanying this submission.

3.5 Minority-class abandonment in DermaMNIST

Returning to RQ1 and RQ2, total collapse is not the only way a pruned classifier can fail, and DermaMNIST illustrates this. It shows little total collapse (Table 5), but per-class recall at 25% data (a mid-range fraction where the dataset's overall Balanced Accuracy is neither floor nor ceiling) reveals a distinct partial failure mode. Table 10 reports all seven classes with standard deviation across 5 seeds; test-set class counts are: akiec 66, bcc 103, bkl 220, df 23, mel 223, nv 1341, vasc 29 (from the official MedMNIST v2 test split). Applying the pre-specified abandonment criterion (recall < 0.05; Methods, 'Metrics and collapse definitions') uniformly to all seven classes shows that only *df* (dermatofibroma, the rarest class by test count) meets it, at every sparsity level shown, including 0% sparsity (mean recall 0.009 ± 0.017 at baseline, exactly 0.000 ± 0.000 at 70%, 90%, and 95% sparsity across all 5 seeds). No other class crosses the threshold at any tested sparsity. This conclusion is not an artefact of the 0.05 cutoff: at thresholds of 0.01 and 0.10 *df* is still the only class with a mean recall below the threshold at 0%, 70% and 90% sparsity, and at 95% sparsity with a threshold of 0.01. Only the loosest combination (0.10 at 95% sparsity) adds *akiec* (0.070) and *vasc* (0.062), each of which falls below 0.05 in 4/5 and 3/5 seeds respectively at that sparsity (Table 10). With 23 test images, a recall of 0/23 only bounds the true recall of *df* below about 0.12 (one-sided 95% confidence), so we describe abandonment as a failure to detect the class in this test set, not as an exact zero recall. *df*'s abandonment is present independent of pruning at 0% sparsity, on only 23 test images to begin with; pruning's contribution is moving mean recall from already below threshold to exactly zero with zero cross-seed variance, not causing the abandonment itself. This failure is invisible to the total-collapse criterion and to AUC-macro, which stays above 0.75 throughout.

Table 10 Per-class recall on DermaMNIST at 25% training data (ResNet-18, adaptive protocol) by sparsity; mean ± population standard deviation over five seeds.

Sp.	akiec	bcc	bkl	df	mel	nv	vasc
0%	.139±.134	.384±.054	.465±.099	**.009±.017**	.322±.193	.862±.053	.228±.210
70%	.142±.091	.431±.142	.495±.146	**.000±.000**	.208±.143	.888±.024	.117±.083
90%	.221±.105	.340±.104	.470±.116	**.000±.000**	.167±.127	.906±.048	.152±.146
95%	.070±.118	.489±.044	.359±.054	**.000±.000**	.152±.078	.926±.022	.062±.094

3.6 Cross-architecture comparison

We turn now to RQ4, beginning with network depth. We compared the three architectures at 90% sparsity in the adaptive configuration, at each dataset's lowest tested fraction, with matched fraction, sparsity, seed count (5), and pipeline (Fig. 4; Balanced Accuracy throughout, consistent with our primary metric). No dataset shows ResNet-50 as the most sparsity-tolerant architecture. ResNet-34 is numerically highest on Pneumonia and Blood; ResNet-18 is numerically highest on Derma, Organs, and Retina. Differences between architectures at any given dataset are typically 1-4

percentage points (ResNet-50 on BloodMNIST is about 8 points below ResNet-18), which we describe as numerical differences, not statistically supported ones: we have not computed paired significance tests across architectures, and Fig. 4 does not include error bars in its current form. Both are noted as scope limitations (see 'Limitations' in the Discussion). We also did not match architectures by parameter count or MACs at a given nominal sparsity; ResNet-50 retains more parameters than ResNet-18 at the same nominal sparsity percentage, so this comparison is at matched percentage sparsity, not matched absolute model size.

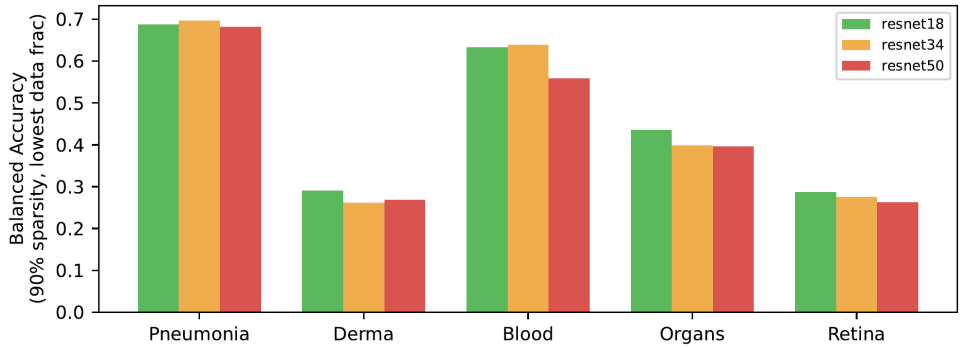

Fig. 4 Balanced Accuracy at 90% sparsity across architectures.

3.7 ResNet-50 under the fixed protocol

The cross-architecture comparison above uses only the adaptive configuration for ResNet-50, so we separately ran a fixed-configuration ResNet-50 condition to test whether the fixed-protocol collapse pattern also holds at this depth. This condition was run at 95% sparsity only (not aggregated across 70-95%), at each dataset's lowest tested fraction, with 5 seeds (Table 11). ResNet-50 collapses in 5/5 seeds on Pneumonia, Blood, and Retina, the same three datasets where ResNet-18 also collapses under the fixed configuration, and in 0/5 on Derma, again matching the smaller architectures. On OrganSMNIST, ResNet-50 collapses in 3/5 seeds at this specific cell, numerically higher than the 0/5 (ResNet-18, at 95% specifically, per Table 5); we report no fixed-protocol rates for ResNet-34. We describe this as numerically higher without a supporting statistical test, since we have not run a significance test comparing collapse rates across architectures at this single cell.

3.8 Pruning criterion

For the pruning-criterion element of RQ4, we compared L1 against random filter selection at a single cell (95% sparsity, each dataset's lowest tested fraction), 5 seeds, ResNet-18 (Table 12, Fig. 5). Both criteria were run with the fixed-batch protocol (batch 128, 75 fine-tuning epochs with early stopping on validation AUC, patience 8)

Table 11 ResNet-50 fixed-protocol results at 95% sparsity.

Dataset (frac)	Collapse	Mean BA
Pneumonia (1%)	5/5	0.500
Blood (1%)	5/5	0.125
Organs (1%)	3/5	0.095
Retina (10%)	5/5	0.200
Derma (10%)	0/5	0.266

and with the pruning ratio found by bisection so that the measured parameter sparsity is 0.95 (a CPU rerun with the code in the release). In this regime both criteria collapse on BloodMNIST (5/5 seeds each) and RetinaMNIST (5/5 for L1, 3/5 for random selection), and often on PneumoniaMNIST (2/5 and 4/5); neither collapses on DermaMNIST or OrganSMNIST, so only these two datasets compare the criteria on models that still predict several classes. The sign of the mean difference changes across datasets ($+2.5$, -3.0, 0.0, $+5.2$ and -1.3 points for Pneumonia, Derma, Blood, Organs and Retina). Paired Wilcoxon signed-rank tests over seeds give $p \geq 0.18$ before correction and $p \geq 0.31$ after Benjamini-Hochberg correction across the five datasets; with five pairs the smallest attainable two-sided p is 0.0625, so the test has little power. We therefore find no evidence for a consistent advantage of L1 over random selection, but cannot claim equivalence either.

Table 12 L1 versus random filter pruning at 95% sparsity (ResNet-18, fixed-batch protocol, 5 seeds): test Balanced Accuracy (mean ± sample SD), number of collapsed seeds, and paired Wilcoxon p (Benjamini-Hochberg adjusted across datasets).

Dataset (frac)	L1 BA	Random BA	Diff. (pp)	Collapsed L1, random	p_{BH}
Pneumonia (1%)	$0.557 \pm .067$	$0.532 \pm .071$	$+2.5$	2/5, 4/5	0.74
Derma (10%)	$0.244 \pm .050$	$0.275 \pm .020$	-3.0	0/5, 0/5	0.31
Blood (1%)	$0.125 \pm .000$	$0.125 \pm .000$	0.0	5/5, 5/5	1.00
Organs (1%)	$0.377 \pm .017$	$0.325 \pm .086$	$+5.2$	0/5, 0/5	0.31
Retina (10%)	$0.200 \pm .000$	$0.213 \pm .018$	-1.3	5/5, 3/5	0.31

3.9 Step-count, BatchNorm and optimizer controls

Because the fixed and adaptive protocols differ in several variables at once, this subsection asks which individual post-pruning choices are sufficient to prevent collapse (RQ3) and whether the adaptive pattern holds across optimizer families (RQ4). We ran a combined set of fine-tuning-only controls from the same trained-and-pruned checkpoint per seed, at each dataset's own lowest tested fraction and 95% sparsity, across PneumoniaMNIST, BloodMNIST, and OrganSMNIST (all three architectures) and, separately, DermaMNIST and RetinaMNIST at their 10% lowest fraction (ResNet-18/34/50, combined-freeze BatchNorm variant only; the decomposed BatchNorm variants below

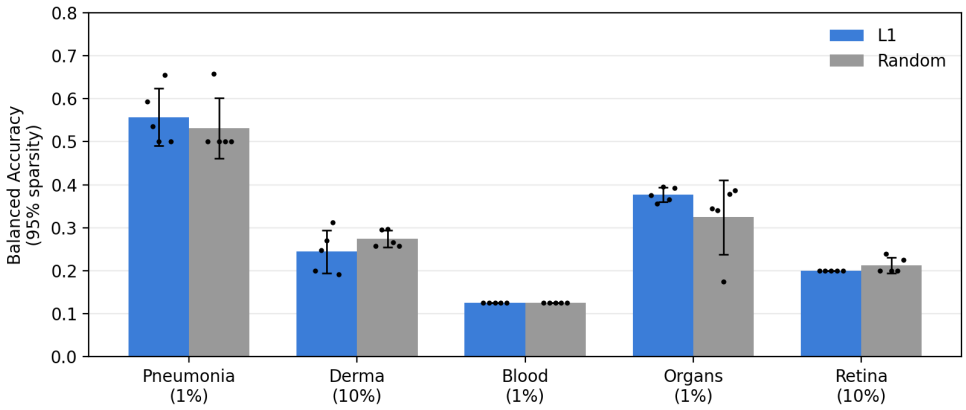

Fig. 5 Balanced Accuracy for L1 and random filter pruning at 95% sparsity. Bars show the mean, error bars the sample SD, and dots the five individual seeds.

were run on the first three datasets only, a scope limitation noted in 'Limitations' in the Discussion). Table 14 and Fig. 6 summarize the collapsed-seed counts.

Step-count-only (batch=128, unscaled base learning rate, epochs raised to match the adaptive condition's total step count for that cell, e.g. 75 epochs for PneumoniaMNIST at 1% data, computed per cell as described in Methods, 'Architectures and training protocols'): 0/5 seeds degenerate on Pneumonia, Blood, Organs, and Retina in every architecture tested (Table 14). The one exception is DermaMNIST with ResNet-34, where 1 of 5 seeds collapsed (1/15 in a 15-seed repeat). Together with the LR-only control (Results, 'Learning-rate scaling at matched step count'), this shows that step count alone, with no change to batch size or learning rate, prevents collapse in all but that one tested cell, including RetinaMNIST.

BatchNorm, decomposed (Pneumonia, Blood, Organs only, Table 13): we separately froze (a) running statistics only, letting affine parameters continue training; (b) affine parameters only, letting running statistics continue updating; (c) both, matching our original single frozen-BN condition; and (d) recalibrated running statistics via forward passes with no gradient updates, then froze them before fine-tuning. Freezing running statistics, alone (stats-only) or combined with affine parameters (both), reintroduces collapse or near-collapse on Pneumonia (5/5 for both variants, ResNet-18) and partial collapse on Blood (3/5 stats-only, 2/5 both) and Organs (1/5 for both variants). Freezing *only* the affine parameters, or recalibrating statistics without gradients, does not reintroduce collapse on any of the three datasets (0/5 throughout, Balanced Accuracy close to the fully-adaptive condition). The evidence therefore implicates frozen running statistics specifically, rather than BatchNorm as a whole or frozen affine parameters.

Optimizer generalization (SGD+momentum, LARS, LAMB, each under both fixed and adaptive batch/LR settings; full hyperparameters: SGD momentum 0.9, weight decay 5×10^{-4} matching AdamW's, no per-optimizer learning-rate search; all three use the same base learning rate and the same 128/eff_batch rule as AdamW,

Table 13 BatchNorm ablation: Balanced Accuracy and collapse count.

Dataset	Stats-only	Affine-only	Both	Recalibrate
Pneumonia	0.500 (5/5)	0.708 (0/5)	0.500 (5/5)	0.716 (0/5)
Blood	0.206 (3/5)	0.655 (0/5)	0.249 (2/5)	0.625 (0/5)
Organs	0.164 (1/5)	0.395 (0/5)	0.180 (1/5)	0.394 (0/5)

which we did not separately verify is optimal for each optimizer, as noted in 'Limitations' in the Discussion): every optimizer collapses under its fixed condition on Pneumonia and Blood (5/5 seeds, matching AdamW) and largely recovers under its adaptive condition (0-1/5 seeds; ResNet-34/LARS-adaptive on Organs is the one exception at 1/5). On Organs, where AdamW's fixed condition does not collapse, SGD and LAMB do not either and LARS collapses in 2/5 seeds. LAMB is reported in the text only: 5/5 fixed and 0/5 adaptive collapse on Pneumonia and Blood, and 0/5 on Organs in both conditions. On DermaMNIST and RetinaMNIST the pattern is less clean: SGD and LARS show partial collapse under *both* fixed and adaptive conditions on Derma (SGD-adaptive 3/5, LARS-fixed 1/5, LARS-adaptive 1/5) and Retina (SGD-fixed 5/5, LARS-fixed 3/5), unlike Pneumonia/Blood/Organs' cleaner fixed-collapses/adaptive-recovers pattern. Optimizer choice thus interacts with dataset in a way our current experiments do not fully explain. We report this next to the cleaner three-dataset result and list it under 'Future work' in the Discussion rather than folding it into the general finding.

Table 14 Collapsed seeds under step-count and optimizer controls.

Dataset	Fix.	Ad.	Step	fBN	SGD-f	SGD-a	LARS-f	LARS-a
Pneumonia	5	0	0	5	5	1	5	0
Blood	5	0	0	2	5	0	5	0
Organs	0	0	0	1	0	0	2	0
Derma	0	0	0	5	2	3	1	1
Retina	3	0	0	4	5	0	3	0

3.10 Batch-size sensitivity

A last check on RQ3 asks whether the adaptive batch size itself matters. On PneumoniaMNIST, 1% data, 95% sparsity, we swept fine-tuning batch size directly with the learning rate set to $10^{-3} \cdot 128/\text{batch}$ at each size. A first sweep used 5 seeds at batch sizes 8, 16, 24, 32 and 64 (Table 15). Mean Balanced Accuracy stays in a 0.57-0.66 band, but every batch size shows 1-2 of 5 seeds degenerate, which disagrees with the complete (0/5) recovery of the standard adaptive condition, which uses eff_batch = 11 (Results, 'Fixed-protocol collapse by sparsity level'; Table 7). A later repeat with 15 seeds, which adds batch size 11, found no degenerate seed at any batch size (0/15 each;

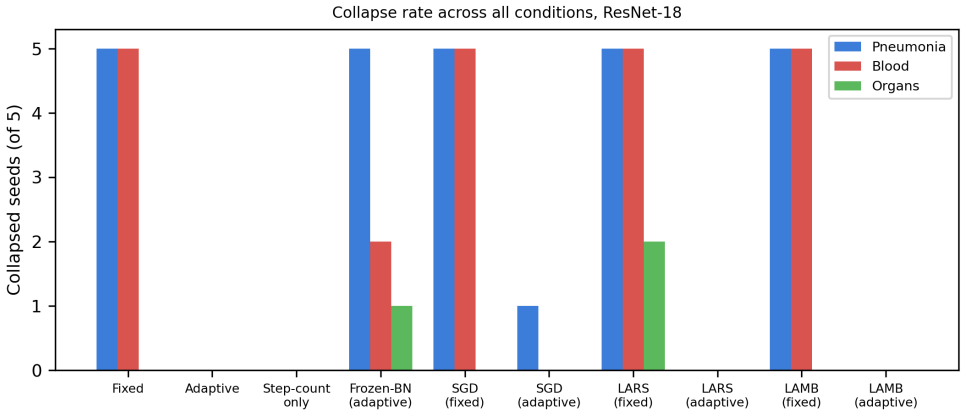

Fig. 6 Collapsed seeds across ten conditions and three datasets.

Wilson 95% upper bound 0.20). We give more weight to the 15-seed repeat because it has three times the seeds and covers the heuristic's own batch size. Its code sets the number of epochs so that every batch size receives at least 75 updates; the log of the 5-seed sweep does not record the epoch count, so we cannot confirm that it used the same update budget. A smaller budget in the first sweep would be consistent with the step-count control above, but our data do not demonstrate it. Because batch size and learning rate change together in both sweeps, they test the joint rule at each size, not batch size in isolation.

Table 15 Batch-size sensitivity on PneumoniaMNIST at 1% data and 95% sparsity: first 5-seed sweep, and the 15-seed repeat (counts of degenerate seeds). Dashes: not run in the 5-seed sweep. The 5-seed file stores no predicted-label counts, so a seed is counted as degenerate when its Balanced Accuracy is exactly 0.5 and its MCC is 0.

Batch size	8	11	16	24	32	64
Mean BA (5 seeds)	0.637	-	0.568	0.639	0.662	0.633
Degenerate (of 5)	1	-	2	2	1	2
Degenerate (of 15)	0	0	0	0	0	0

4 Discussion

We asked whether one-shot structured pruning can hide collapse under whole-image medical data scarcity. For RQ1, the fixed protocol collapsed on three of five datasets (PneumoniaMNIST, BloodMNIST and RetinaMNIST), with a rate that varied by dataset and sparsity level, whereas the adaptive protocol eliminated collapse on four

datasets and reduced it on RetinaMNIST (Results, 'Fixed-protocol collapse by sparsity level'). For RQ2, AUC and Accuracy remained higher than Balanced Accuracy and MCC during collapse, so class-sensitive metrics are required (Results, 'Metric masking'). RQ3 and RQ4 are addressed in the paragraphs below.

The results of the sparsity comparison, the learning-rate control and the step-count, BatchNorm and optimizer controls (Results, 'Fixed-protocol collapse by sparsity level', 'Learning-rate scaling at matched step count' and 'Step-count, BatchNorm and optimizer controls') support a limited causal statement for RQ3: aggressive pruning, severe data scarcity, and insufficient post-pruning adaptation occur together with collapse on three of five datasets, and increasing step count, scaling the learning rate, or adapting BatchNorm running statistics each prevented it in most of the cells we tested, on its own (learning-rate scaling alone left 1/5 BloodMNIST and 2/5 RetinaMNIST seeds collapsed, and the step-count control left 1/5 DermaMNIST ResNet-34 seeds collapsed). This is evidence of an interaction among these factors, not proof of a single cause. We do not claim any one factor is individually necessary, only that each tested intervention reduced collapse at the cells we tested. Describing the fixed configuration's failure as caused by "batch size alone" would be incorrect, since the fixed configuration also changes step count and BatchNorm update dynamics relative to adaptive, so we avoid that phrasing.

The BatchNorm decomposition (Results, 'Step-count, BatchNorm and optimizer controls') is best read as evidence that post-pruning BatchNorm *adaptation* matters, in that running statistics need to update during fine-tuning, and not as evidence that BatchNorm statistics computed on a small batch are intrinsically harmful; recalibrating those statistics without any gradient training was sufficient to avoid collapse, which suggests the issue is stale statistics rather than statistics computed on small batches per se. We do not claim this factor is "at least as important" as learning rate or step count; we have not run a comparable effect-size analysis across the three interventions and do not have the evidence to rank them.

Two datasets (DermaMNIST, OrganSMNIST) show little total collapse under the fixed configuration at the fractions tested. For DermaMNIST, total collapse is not the relevant failure mode at all: a minority-class recall failure (Results, 'Minority-class abandonment in DermaMNIST') exists independently of total collapse, already present at 0% sparsity on a class with only 23 test images.

For RQ4, the patterns are only partly stable. The cross-architecture (Results, 'Cross-architecture comparison') and pruning-criterion (Results, 'Pruning criterion') results are mixed: no architecture is consistently most tolerant, and the sign of the L1-minus-random difference changes across datasets and no paired test is significant after correction, so we find no consistent L1 advantage at this seed count. The fixed-collapses and adaptive-recovers pattern was reproduced under SGD, LARS and LAMB on Pneumonia, Blood and Organs but was less clean on Derma and Retina, so the adaptive schedule does not transfer uniformly across optimizer families (Results, 'Step-count, BatchNorm and optimizer controls').

5 Limitations

Statistical treatment of repeated observations. Our primary tables report per-sparsity-level results (Table 5) rather than only a pooled $n = 20$ statistic, but we have not fit a mixed-effects or repeated-measures model with seed as a random effect, which would be the more rigorous treatment of the fact that the same seed's baseline checkpoint and training subsample are shared across sparsity levels. The interaction regression (Results, 'Interaction regression') uses ordinary OLS on a bounded $[0, 1]$ outcome with repeated seeds; we do not correct for either issue.

Sample size and variance. Most conditions use 5 seeds per cell; several comparisons (L1-vs-random, cross-architecture) show standard deviations large enough that we do not treat mean differences under roughly 10 percentage points as confirmed effects without a supporting test we have not run.

Scope not yet extended. The decomposed BatchNorm ablation (stats-only, affine-only, recalibrate) is run on Pneumonia, Blood, and Organs only; Derma and Retina have only the combined-freeze variant. The optimizer-generalization controls show a more complex, only partially understood pattern on Derma and Retina (Table 14) that we have not explained. The partial-collapse (class-abandonment) analysis is applied to DermaMNIST only. We use only one-shot L1/random pruning, not iterative schedules, non-magnitude criteria (Taylor, Gate Decorator), or compensation-based one-shot methods that rebuild the retained channels in closed form [37]. Such compensation, together with BatchNorm re-estimation after pruning [38, 39] and data-free recovery that matches stored BatchNorm statistics [40], are alternative ways to avoid collapse that our protocol-level controls do not compare against. Collapse is defined as a single predicted label; we did not compute continuous indicators such as the entropy or Gini index of the predicted-label distribution because the per-run logs do not store predictions.

Training details not fully reported. We used PyTorch's default AdamW betas (0.9, 0.999); we did not apply gradient clipping; loss is standard cross-entropy with no class weighting; the CosineAnnealingLR scheduler steps once per epoch. The software versions of the original GPU runs were not recorded (the CPU rerun of the pruning-criterion control used PyTorch 2.11 and Torch-Pruning 1.6.0, and we found that the meaning of the raw pruning ratio changed between Torch-Pruning versions), so exact reproduction of the earlier numbers is not guaranteed, and we did not verify whether optimizer/scheduler state is reset after pruning versus reinitialized fresh (our implementation reinitializes a fresh optimizer for each fine-tuning run, but we have not verified that this holds identically across every notebook version used in the experiments).

Other. No DeLong test (per-sample predictions not retained). No data augmentation or class-weighted/focal-loss training. No external clinical validation set; all results are on MedMNIST v2, a benchmark, not clinical data, at 28×28 resolution. We report parameter and multiply-accumulate reductions (Table 4) but no real-hardware latency or energy measurement. No calibration analysis (ECE is logged but not systematically analyzed in this study; calibration is a distinct evaluation concern [47]). Achieved-sparsity calibration (Table 4) is reported for ResNet-18 only.

6 Future work

In priority order: (1) explain the difference between the 5-seed and 15-seed batch-size sweeps (Results, 'Batch-size sensitivity') by rerunning both with identical update budgets, and run a $2 \times 2 \times 2$ factorial design (learning-rate scaling $\times$ update count $\times$ BatchNorm-statistic update) to estimate main effects and interactions rather than sufficiency alone; (2) extend the decomposed BatchNorm ablation (stats-only, affine-only, recalibrate) to DermaMNIST and RetinaMNIST; (3) investigate the more complex optimizer-generalization pattern observed on Derma and Retina (Table 14); (4) fit a mixed-effects model with seed as a random effect for the collapse-rate and Balanced-Accuracy comparisons; (5) extend the partial-collapse (recall$<$ 0.05) criterion to Blood, Organs, and Retina; (6) add a non-magnitude pruning criterion (Taylor/Gate Decorator) and an iterative pruning schedule at $\geq$90% sparsity, compare against closed-form compensation and BatchNorm re-estimation baselines, and use more seeds for the L1-vs-random comparison; (7) test GroupNorm and higher-resolution datasets, optimizer-specific tuning, and store predicted labels so that continuous collapse indicators (entropy, Gini index) can be computed; (8) real-device latency/energy measurement and calibration analysis; (9) retain per-sample predictions for a proper DeLong test.

7 Conclusions

Across five MedMNIST v2 datasets, a training configuration with a fixed, dataset-size-independent batch size is associated with classifier collapse under extreme sparsity and scarcity on three of five datasets. The most cautious causal reading our evidence supports is that collapse results from an interaction between aggressive pruning, severe data scarcity, and insufficient post-pruning adaptation (through step count, learning rate, and BatchNorm running-statistic updates, each reducing it where we tested), rather than from fixed batch size alone, or from sparsity alone. The main experimental grid uses ResNet-18 and ResNet-34; ResNet-50 is used in additional, separately-scoped experiments (cross-architecture comparison and the step-count/BatchNorm/optimizer controls), not the main grid. Common aggregate metrics (Accuracy, AUC) may not directly reveal hard-label collapse, because they reflect score ranking or class-prior bias rather than the final predicted label; Balanced Accuracy and MCC do reveal it in our data. We find no consistent cross-architecture capacity advantage and no uniformly confirmed L1-vs-random pruning advantage. We report negative, mixed, and only partially understood findings together with the tested scope and the statistical caveats listed under 'Limitations'. The work is a benchmark-level result and needs further validation before any clinical implication is drawn.

Declarations

Data availability. The datasets analyzed in this study were obtained from MedMNIST v2, a publicly available benchmark for biomedical image classification, accessible at https://medmnist.com/.

Code availability. The source code developed for this study, including model training, structured pruning, post-pruning optimization, evaluation procedures, and statistical analysis scripts, is available from the corresponding author upon reasonable request.

Acknowledgements. The authors gratefully acknowledge Universidad Europea del Atlántico, Santander, Spain, for its financial support of this research. The authors also acknowledge the developers and contributors of the MedMNIST v2 benchmark for making the biomedical imaging datasets publicly accessible for scientific research.

Author contributions. Z.A. contributed to conceptualization, methodology, software development, validation, formal analysis, investigation, data curation, visualization, and writing the original draft.

I.d.l.T.D. contributed to supervision, project administration, and writing-review and editing.

P.C. contributed to methodology, validation, and writing-review and editing.

A.M.H.C. contributed to investigation, validation, and writing-review and editing.

Y.M.N. contributed to data curation, formal analysis, and writing-review and editing.

M.Z.J. contributed to conceptualization, methodology, supervision, and writing-review and editing.

H.M.R.u.R. contributed to conceptualization, methodology, supervision, project administration, and writing-review and editing.

All authors reviewed and approved the final version of the manuscript.

Competing interests. The authors declare that they have no competing interests.

Funding. This research was financially supported by Universidad Europea del Atlántico, Santander, Spain.

References

[1] Han, S., Pool, J., Tran, J. & Dally, W. J. Learning both weights and connections for efficient neural networks. in *Advances in Neural Information Processing Systems* Vol. 28, 1135-1143 (2015). https://arxiv.org/abs/1506.02626

[2] Wen, W., Wu, C., Wang, Y., Chen, Y. & Li, H. Learning structured sparsity in deep neural networks. in *Advances in Neural Information Processing Systems* Vol. 29, 2074-2082 (2016). https://arxiv.org/abs/1608.03665

[3] Li, H., Kadav, A., Durdanovic, I., Samet, H. & Graf, H. P. Pruning filters for efficient ConvNets. in *Proc. Int. Conf. Learn. Represent.* (2017). https://arxiv.org/abs/1608.08710

[4] Liu, Z. et al. Learning efficient convolutional networks through network slimming. in *Proc. IEEE Int. Conf. Comput. Vis.* 2755-2763 (2017). https://doi.org/10.1109/ICCV.2017.298.

[5] Dinsdale, N. K., Jenkinson, M. & Namburete, A. I. L. STAMP: simultaneous training and model pruning for low data regimes in medical image segmentation. *Med. Image Anal.* **81**, 102583 (2022). https://doi.org/10.1016/j.media.2022.102583.

[6] Götz, T. I. et al. Number of necessary training examples for neural networks with different number of trainable parameters. *J. Pathol. Inform.* **13**, 100114 (2022). https://doi.org/10.1016/j.jpi.2022.100114.

[7] Holste, G. et al. How does pruning impact long-tailed multi-label medical image classifiers? in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2023* (Lecture Notes in Computer Science Vol. 14224) 663-673 (Springer, 2023). https://doi.org/10.1007/978-3-031-43904-9_64.

[8] Larsson, G. & Odin, M. The impact of pruning convolutional neural networks when classifying skin cancer. Bachelor's thesis, KTH Royal Institute of Technology (2023).

[9] Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. in *Proc. Int. Conf. Mach. Learn.* 448-456 (2015). https://arxiv.org/abs/1502.03167

[10] Douzandeh Zenoozi, A., Erhan, L., Liotta, A. & Cavallaro, L. A comparative study of neural network pruning strategies for industrial applications. *Front. Comput. Sci.* **7**, 1563942 (2025). https://doi.org/10.3389/fcomp.2025.1563942.

[11] Banerjee, T. et al. A novel unified Inception-U-Net hybrid gravitational optimization model (UIGO) incorporating automated medical image segmentation and feature selection for liver tumor detection. *Sci. Rep.* **15**, 29908 (2025). https://doi.org/10.1038/s41598-025-14333-0.

[12] de Paula, E. R., Soni, J., Upadhyay, H. & Lagos, L. Comparative analysis of model compression techniques for achieving carbon efficient AI. *Sci. Rep.* **15**, 23461 (2025). https://doi.org/10.1038/s41598-025-07821-w.

[13] Abd El-Hafeez, T., Tarek, M. & Sayed, A. Optimizing YOLOv11 for automated classification of breast cancer in medical images. *Sci. Rep.* **15**, 37170 (2025). https://doi.org/10.1038/s41598-025-24850-7.

[14] Saha, S. & Xu, L. Vision transformers on the edge: a comprehensive survey of model compression and acceleration strategies. *Neurocomputing* **643**, 130417 (2025). https://doi.org/10.1016/j.neucom.2025.130417.

[15] Liu, S., Yue, W., Guo, Z. & Wang, L. Multi-branch CNN and grouping cascade attention for medical image classification. *Sci. Rep.* **14**, 15013 (2024). https://doi.org/10.1038/s41598-024-64982-w.

[16] Dheivya, I. & Kumar, G. S. VSegNet - a variant SegNet for improving segmentation accuracy in medical images with class imbalance and limited data. *Medinformatics* **2**, 36-48 (2024). https://doi.org/10.47852/bonviewmedin42023518.

[17] Ren, K., Hong, G., Chen, X. & Wang, Z. A COVID-19 medical image classification algorithm based on Transformer. *Sci. Rep.* **13**, 5359 (2023). https://doi.org/10.1038/s41598-023-32462-2.

[18] Cheng, H., Zhang, M. & Shi, J. Q. A survey on deep neural network pruning: taxonomy, comparison, analysis, and recommendations. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 10558-10578 (2024). https://doi.org/10.1109/TPAMI.2024.3447085.

[19] Roschewitz, M. et al. Automatic correction of performance drift under acquisition shift in medical image classification. *Nat. Commun.* **14**, 6608 (2023). https://doi.org/10.1038/s41467-023-42396-y.

[20] Wang, C. Calibration in deep learning: a survey of the state-of-the-art. Preprint at https://arxiv.org/abs/2308.01222 (2023).

[21] Adamczewski, K. & Park, M. Differential privacy meets neural network pruning. Preprint at https://arxiv.org/abs/2303.04612 (2023).

[22] Benbaki, R. et al. Fast as CHITA: neural network pruning with combinatorial optimization. Preprint at https://arxiv.org/abs/2302.14623 (2023).

[23] Li, M., Jiang, Y., Zhang, Y. & Zhu, H. Medical image analysis using deep learning algorithms. *Front. Public Health* **11**, 1273253 (2023). https://doi.org/10.3389/fpubh.2023.1273253.

[24] Rafat, K. et al. Mitigating carbon footprint for knowledge distillation based deep learning model compression. *PLoS ONE* **18**, e0285668 (2023). https://doi.org/10.1371/journal.pone.0285668.

[25] Jani, M., Fayyad, J., Al-Younes, Y. & Najjaran, H. Model compression methods for YOLOv5: a review. Preprint at https://arxiv.org/abs/2307.11904 (2023).

[26] Zhang, Z., Tao, R. & Zhang, J. Neural network pruning by gradient descent. Preprint at https://arxiv.org/abs/2311.12526 (2023).

[27] Huang, S.-C. et al. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *npj Digit. Med.* **6**, 74 (2023). https://doi.org/10.1038/s41746-023-00811-0.

[28] Sambyal, A. S., Niyaz, U., Krishnan, N. C. & Bathula, D. R. Understanding calibration of deep neural networks for medical image classification. *Comput. Methods Programs Biomed.* **242**, 107816 (2023). https://doi.org/10.1016/j.cmpb.2023.107816.

[29] Wang, H., Qin, C., Bai, Y. & Fu, Y. Why is the state of neural network pruning so confusing? On the fairness, comparison setup, and trainability in network pruning. Preprint at https://arxiv.org/abs/2301.05219 (2023).

[30] Ansari, M. Y. et al. A lightweight neural network with multiscale feature enhancement for liver CT segmentation. *Sci. Rep.* **12**, 14153 (2022). https://doi.org/10.1038/s41598-022-16828-6.

[31] Song, Z. et al. CP-ViT: cascade vision transformer pruning via progressive sparsity prediction. Preprint at https://arxiv.org/abs/2203.04570 (2022).

[32] Rajaraman, S., Ganesan, P. & Antani, S. Deep learning model calibration for improving performance in class-imbalanced medical image classification tasks. *PLoS ONE* **17**, e0262838 (2022). https://doi.org/10.1371/journal.pone.0262838.

[33] Gale, T., Elsen, E. & Hooker, S. The state of sparsity in deep neural networks. Preprint at https://arxiv.org/abs/1902.09574 (2019).

[34] Liu, Z., Sun, M., Zhou, T., Huang, G. & Darrell, T. Rethinking the value of network pruning. in *Proc. Int. Conf. Learn. Represent.* (2019). https://arxiv.org/abs/1810.05270.

[35] Blalock, D., Gonzalez Ortiz, J. J., Frankle, J. & Guttag, J. What is the state of neural network pruning? in *Proc. Mach. Learn. Syst.* (2020). https://arxiv.org/abs/2003.03033

[36] Fang, G., Ma, X., Song, M., Mi, M. B. & Wang, X. DepGraph: towards any structural pruning. in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 16091-16101 (2023). https://arxiv.org/abs/2301.12900

[37] Zhang, B. & Yang, B. CORP: closed-form one-shot representation-preserving structured pruning for transformers. Preprint at https://arxiv.org/abs/2602.05243 (2026).

[38] Li, Y., Wang, N., Shi, J., Liu, J. & Hou, X. Revisiting batch normalization for practical domain adaptation. Preprint at https://arxiv.org/abs/1603.04779 (2016).

[39] Li, B., Wu, B., Su, J., Wang, G. & Lin, L. EagleEye: fast sub-net evaluation for efficient neural network pruning. in *Computer Vision - ECCV 2020* 639-654 (Springer, 2020). https://doi.org/10.1007/978-3-030-58536-5_38

[40] Yin, H. et al. Dreaming to distill: data-free knowledge transfer via DeepInversion. in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 8712-8721 (2020). https://doi.org/10.1109/CVPR42600.2020.00874

[41] Yang, J. et al. MedMNIST v2 - a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci. Data* **10**, 41 (2023). https://doi.org/10.1038/s41597-022-01721-8.

[42] Goyal, P. et al. Accurate, large minibatch SGD: training ImageNet in 1 hour. Preprint at https://arxiv.org/abs/1706.02677 (2017).

[43] You, Y., Gitman, I. & Ginsburg, B. Large batch training of convolutional networks. Preprint at https://arxiv.org/abs/1708.03888 (2017).

[44] You, Y. et al. Large batch optimization for deep learning: training BERT in 76 minutes. in *Proc. Int. Conf. Learn. Represent.* (2020). https://arxiv.org/abs/1904.00962.

[45] Malladi, S., Lyu, K., Panigrahi, A. & Arora, S. On the SDEs and scaling rules for adaptive gradient algorithms. in *Advances in Neural Information Processing Systems* Vol. 35 (2022). https://arxiv.org/abs/2205.10287

[46] Marek, M., Lotfi, S., Somasundaram, A., Wilson, A. G. & Goldblum, M. Small batch size training for language models: when vanilla SGD works, and why gradient accumulation is wasteful. Preprint at https://arxiv.org/abs/2507.07101 (2025).

[47] Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. in *Proc. Int. Conf. Mach. Learn.* 1321-1330 (2017). https://arxiv.org/abs/1706.04599

[48] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 770-778 (2016). https://arxiv.org/abs/1512.03385

[49] Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **10**, e0118432 (2015). https://doi.org/10.1371/journal.pone.0118432

[50] Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020). https://doi.org/10.1186/s12864-019-6413-7