WHAT DOES RECOVERY RECOVER? AN EMPIRICAL STUDY OF PREDICTION RETENTION AFTER STRUCTURED PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

To reduce the environmental impact of artificial intelligence, researchers aim to reduce the computation required for model inference while maintaining task performance. A common approach uses structured pruning to remove network components and recovery training to regain performance in the compressed model. Recovery is commonly evaluated through overall accuracy. However, improvements on some examples can compensate for unresolved pruning damage on others, concealing persistent failures that affect model reliability. To identify these hidden failures, we investigate how damage repair, corrections of original errors, and losses of correct predictions contribute to recovery gains, and whether repaired predictions remain correct during training. Our empirical study connects overall performance, group outcomes, and individual prediction histories, with matched continuation of the original unpruned model as a training reference. The analysis shows that small overall accuracy differences can coexist with larger group differences as corrections and losses offset each other. Prediction histories reveal the contributions of damage repair and other prediction changes, while accounting for continuation losses can change the interpretation of where additional losses concentrate. Repaired predictions can also become incorrect again. These findings characterize recovery quality through the composition of gains, the distribution of losses, and the persistence of repairs. Based on this analysis, we define a continuation-referenced retention diagnostic that complements overall and group accuracy and negative flips relative to the original model. Together, these measures provide criteria for evaluating recovery training and comparing compressed models according to computational savings, predictive performance, and prediction retention.

1 INTRODUCTION

Green artificial intelligence (AI) emphasizes computational efficiency alongside predictive performance to address the environmental costs of machine learning (Schwartz et al., 2020). To advance this goal, model compression aims to reduce inference computation while maintaining task performance. A common strategy removes network components, such as convolutional channels and their associated parameters, through structured pruning (Li et al., 2017; Liu et al., 2017). Recovery training then optimizes the resulting compressed model to regain performance lost through pruning and meet task requirements (Han et al., 2015; Renda et al., 2020). Its effectiveness is commonly evaluated by overall accuracy (Renda et al., 2020). However, improvements on some examples can offset unresolved pruning damage on others, concealing persistent errors in particular classes or operating contexts. Such errors can undermine compressed-model reliability in safety-critical applications (Liebenwein et al., 2021). For illustration, in a resource-constrained driver-assistance system, recovery training may restore a pruned model's overall accuracy through improved daytime recognition while leaving pruning-induced nighttime pedestrian recognition errors unresolved, creating a potential safety risk. Evaluating recovery therefore requires examining damage repair and prediction retention to identify failures that remain relevant to deployment.

Existing research identifies uneven compression effects across examples and groups (Hooker et al., 2019; Tran et al., 2022), while studies of model updates quantify losses of originally correct pre-

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.

dictions as negative flips (Yan et al., 2021). Recovery training raises a further question about the relationship between pruning damage, subsequent prediction losses, and recovery gains. A final negative-flip rate combines unresolved pruning damage with losses on predictions that initially survive pruning. Recovery gains similarly combine damage repair and corrections of original errors, offset by losses elsewhere. Separating these contributions clarifies the role of damage repair in recovered-model quality. We investigate these relationships by connecting overall and group performance to prediction histories across pruning and recovery. Matched continuation trains the original unpruned model using the same data and update budgets, providing a reference for further optimization (Jin et al., 2022). Our analysis proceeds through three progressively finer research questions. **RQ1. How much performance does recovery regain, and how does its final accuracy compare with the original model and matched continuation? RQ2. How are these performance changes distributed across predefined groups? RQ3. How do repairs, corrections, and losses explain the group outcomes, and do repaired predictions remain correct across saved training checkpoints?**

Our analysis shows that recovery gains reflect both damage repair and changes in the set of correct predictions, with consequences for group performance and prediction retention. Recovered and continuation models attain close mean accuracy in the main comparisons. Group-level analysis reveals larger differences within particular groups and declines from the original model shared by both paths. Individual prediction histories explain these outcomes through corrections that offset losses. A BatchNorm statistics-only diagnostic further shows that high repair within the damage cohort can coexist with losses elsewhere and lower overall accuracy. Subtracting continuation losses can reverse the ordering of loss contributions from the damage and initially spared cohorts, changing the interpretation of where additional losses concentrate. Some repaired predictions also become incorrect again during training. These findings explain recovery gains, loss contributions relative to continued training, and the retention of repaired predictions. We define a continuation-referenced retention diagnostic that measures the fraction of examples correct under both the original and continuation models that the recovered model also classifies correctly. Used alongside overall and group accuracy and negative flips relative to the original model, this diagnostic provides criteria for future research to evaluate recovery training and select compressed models according to computational savings, predictive performance, and prediction retention.

2 Related work

Pruning and recovery training. Weight-pruning methods combine connection removal with recovery training to reduce model size while preserving accuracy (Han et al., 2015). Structured pruning extends this approach to entire filters or channels, producing compact networks with lower computational requirements (Li et al., 2017; Liu et al., 2017). Subsequent research examines the roles of architecture and training. Training compact networks from scratch highlights the importance of architecture (Liu et al., 2019), while weight and learning-rate rewinding demonstrate the influence of training strategies (Renda et al., 2020). Jin et al. (2022) further compare pruning with extended unpruned training under matched steps and learning-rate schedules, relating generalization changes to training-loss differences across examples grouped by estimated influence on generalization.

Prediction changes and group outcomes. Alongside overall performance, compression research examines changes in individual predictions. Pruning Identified Exemplars reveal systematic differences between compressed and uncompressed models with comparable accuracy, exposing disproportionate effects on underrepresented examples (Hooker et al., 2019). Group-level analyses relate disparate pruning effects to gradient norms and distance to the decision boundary (Tran et al., 2022). This focus connects compression evaluation with model-update research, where Yan et al. (2021) quantify negative flips and develop training methods to reduce them alongside overall error. A temporal perspective comes from Toneva et al. (2019), who identify forgetting events in which correctly classified examples become incorrect during training.

These studies motivate examining how damage repair, corrections of original errors, and prediction losses jointly determine recovery quality. We connect these events to overall and group outcomes through prediction histories, with matched continuation as a training reference. This focus links recovery gains to prediction retention and informs criteria for compressed-model comparison.

3 OVERALL PERFORMANCE: RECOVERY RELATIVE TO WHAT?

3.1 A COMMON STARTING MODEL AND TWO MATCHED TRAINING PATHS

Let O be the original model and P the immediately pruned model. Recovery training produces the recovered model C from P. Matched continuation produces the continuation model D from O. Each C, D pair matches example draws, augmentation seed rules, optimizer family, learning-rate schedule, and number of completed parameter updates. We measure computational reduction through operation counts.

For a fixed test set of N inputs x_i with labels y_i, model M has accuracy $\mathrm{Acc}(M) = N^{-1} \sum_i \mathbf{1}[f_M(x_i) = y_i]$. Here f_M predicts the class with the largest logit, and $\mathbf{1}$ is the indicator function. We compare:

$$\underbrace{\mathrm{Acc}(P) - \mathrm{Acc}(O)}_{\text{immediate structural change}}, \qquad \underbrace{\mathrm{Acc}(C) - \mathrm{Acc}(P)}_{\text{recovery gain}}, \qquad \underbrace{\mathrm{Acc}(C) - \mathrm{Acc}(D)}_{\text{matched accuracy difference}}. \tag{1}$$

These comparisons distinguish immediate pruning change, recovery gain, and final accuracy relative to matched continuation.

3.2 EXPERIMENTAL SETTINGS AND AGGREGATION

The nine main settings cross Waterbirds, CelebA, and MetaShift with ResNet-18, DenseNet-121, and ConvNeXt-Tiny (Sagawa et al., 2020; Liu et al., 2015; Liang & Zou, 2022; He et al., 2016; Huang et al., 2017; Liu et al., 2022). Each task has two classes, four metadata-defined groups, and 224×224 inputs. MetaShift uses a fixed random split of its cat/dog contextual pool. Group-sensitive scores estimate channel-removal effects on each group's loss and use the largest estimate. Pruning targets 10% fewer counted floating-point operations (FLOPs), achieving 9.99%–10.05%. Appendix A gives preprocessing and pruning details.

Original models use ImageNet-pretrained weights and group distributionally robust optimization (GroupDRO), which upweights higher-loss groups. Recovery and continuation both optimize cross-entropy. Their paired difference compares the recovery path with matched continuation under this shared objective change, including structure–optimization interactions.

Each setting has three independently trained original models and three paired recovery/continuation repeats per model. We compute paired metrics within repeats, average within each original model, and report mean and standard deviation (SD) across original models. SD measures training variation on a fixed test set.

Both paths use stochastic gradient descent (SGD) with a cosine learning-rate schedule; Appendix A gives hyperparameters. Waterbirds and MetaShift receive 600 updates; CelebA receives 2,500. Fixed endpoints use the final update. Validation-selected endpoints independently maximize each path's worst-group accuracy, the minimum across predefined groups; ties favor overall accuracy, then the earlier update. The smallest selection groups contain 27, 36, and 7 examples for Waterbirds, CelebA, and MetaShift. Both paths train on either the original training set or a reserved validation repair subset; checkpoint selection uses a separate validation subset. The primary comparison uses original training data and fixed endpoints; Section 3.4 examines alternatives.

3.3 RECOVERY GAINS AND MATCHED ACCURACY DIFFERENCES

Across the nine primary settings, mean $\mathrm{Acc}(C) - \mathrm{Acc}(D)$ ranges from -0.470 to $+0.038$ percentage points (pp), with the variation shown in Figure 1 and Table 1. Every absolute mean is below 0.5 pp. Both training paths change accuracy relative to O, with direction and magnitude depending on the setting.

For ResNet-18/Waterbirds, accuracies for O, P, C, D are 88.25%, 86.23%, 86.63%, and 86.73%, respectively. Recovery gains 0.40 ± 2.04 pp from P and ends 0.098 ± 0.021 pp below D. Both endpoints remain below O, placing the recovery gain in the context of shared deterioration.

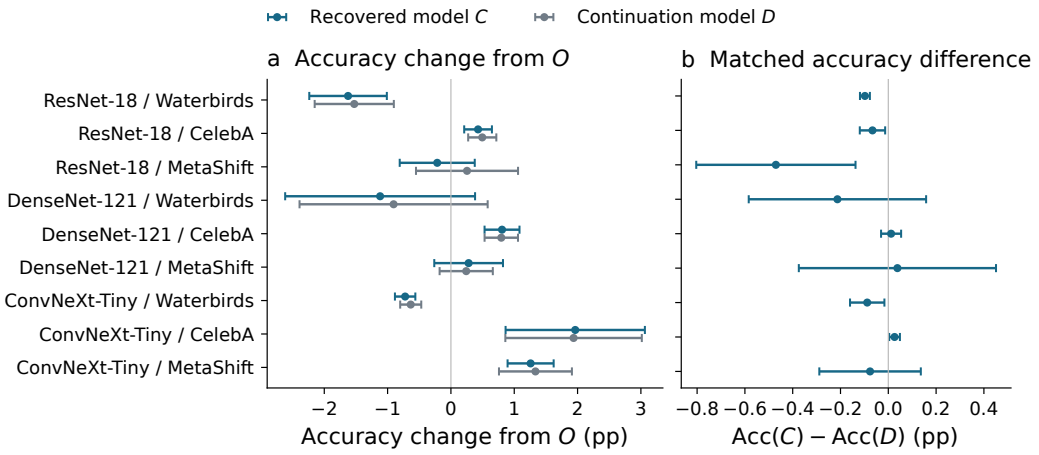


Figure 1: **The training reference changes the interpretation of recovery.** Original-data, fixed-budget results for every architecture and dataset. (a) Accuracy change from the original model for each training path. (b) Their paired difference. Points and bars are mean ± SD across three original models, after averaging three recovery repeats.

Table 1: **Complete main-grid test accuracy.** Percent, except the final column in pp; mean ± SD over three original models, each averaging three recovery repeats. O: original; P: immediately pruned; C: recovered; D: continuation. Original training data and fixed update budgets.

Architecture / dataset	O	P	C	D	$C - D$ (pp)
ResNet-18 / Waterbirds	88.25 ± 0.26	86.23 ± 1.68	86.63 ± 0.41	86.73 ± 0.42	-0.10 ± 0.02
ResNet-18 / CelebA	95.01 ± 0.27	95.04 ± 0.20	95.44 ± 0.18	95.50 ± 0.13	-0.07 ± 0.05
ResNet-18 / MetaShift	87.41 ± 0.79	83.52 ± 1.10	87.20 ± 0.27	87.67 ± 0.50	-0.47 ± 0.33
DenseNet-121 / Waterbirds	91.45 ± 0.96	91.38 ± 0.46	90.33 ± 0.54	90.54 ± 0.60	-0.21 ± 0.37
DenseNet-121 / CelebA	94.79 ± 0.18	94.84 ± 0.11	95.60 ± 0.10	95.59 ± 0.10	0.01 ± 0.04
DenseNet-121 / MetaShift	90.24 ± 0.17	87.99 ± 2.36	90.52 ± 0.41	90.48 ± 0.25	0.04 ± 0.41
ConvNeXt-Tiny / Waterbirds	92.10 ± 0.44	92.18 ± 0.42	91.37 ± 0.28	91.46 ± 0.28	-0.09 ± 0.07
ConvNeXt-Tiny / CelebA	93.85 ± 1.14	93.84 ± 1.18	95.81 ± 0.04	95.79 ± 0.06	0.03 ± 0.02
ConvNeXt-Tiny / MetaShift	91.50 ± 0.26	91.23 ± 0.40	92.75 ± 0.43	92.83 ± 0.61	-0.08 ± 0.21

A complementary CIFAR-100 experiment uses ResNet-18 trained from scratch with cross-entropy, channel pruning ranked by weight magnitude or random order, and 10%/20% FLOP targets (Krizhevsky, 2009). Three original-model seeds each have two recovery repeats with 14,080 updates. Under random 20% pruning, recovery gains 34.07 pp from P and ends 0.52 pp below D (Table 25). This multiclass experiment also supplies the trajectories in Section 5.6.

3.4 Endpoint and Training Data Change the Comparison

We examine how checkpoint selection and training data affect the accuracy comparison. Across the nine original-data settings, validation-selected mean $C - D$ differences range from -1.220 to $+0.054$ pp. For ResNet-18/Waterbirds, the gap changes from -0.098 pp at fixed endpoints to -1.220 pp at selected endpoints. Fixed endpoints compare equal realized updates; selected endpoints compare each path's validation choice within a common maximum budget.

Training data also affect both paths. On the same Waterbirds models, continuation changes accuracy by -1.53 pp with original training data and $+4.50$ pp with the repair subset. These regimes differ in composition and exposure per example. Appendix B gives the paired outcomes for every combination of data and endpoint, and Appendix A specifies the selection-group counts and exposure budgets.

The group and individual analyses use original training data and fixed endpoints unless stated otherwise. RQ2 first asks how groups contribute to the small overall differences.

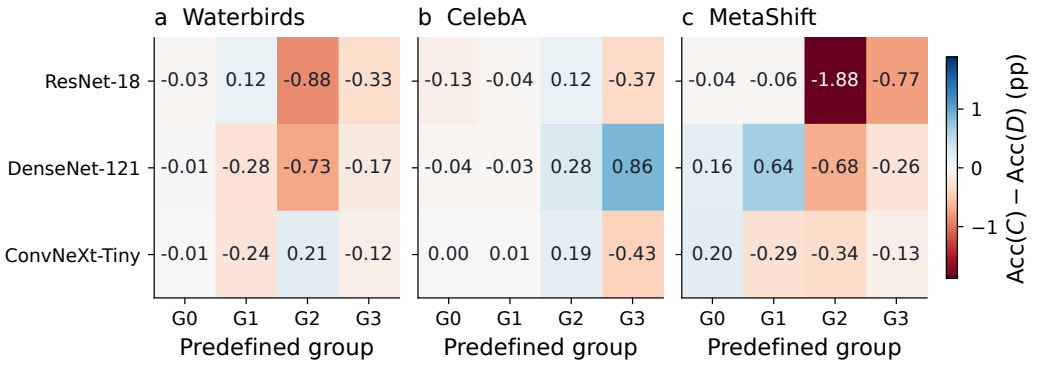


Figure 2: **Group outcomes vary across the evaluated models.** All predefined groups in the original-data, fixed-budget comparison. Cells show mean $C - D$ accuracy differences in pp over three original models, each averaging three recovery repeats. Group $g = 2y + a$ combines binary label y and attribute/context a. Values $y = 1, a = 1$ mean waterbird/water background, blond hair/Male, and cat/indoor, respectively. Group counts and SDs appear in Table 11; panels share a color scale centered at zero.

4 Group outcomes: where do the performance differences occur?

4.1 A complete decomposition over predefined groups

The predefined groups combine task labels with background in Waterbirds, the Male attribute in CelebA, and indoor/outdoor context in MetaShift. For fixed, disjoint groups g, let N_g be group size and $\mathrm{Acc}_g(M)$ group accuracy. Then

$$\mathrm{Acc}(C) - \mathrm{Acc}(D) = \sum_g \frac{N_g}{N} \big[\, \mathrm{Acc}_g(C) - \mathrm{Acc}_g(D) \big]. \tag{2}$$

Each group contributes its accuracy difference weighted by its test proportion; opposite signs offset each other. Figure 2 gives paired differences for every group; Table 11 gives accuracies of O, C, D and group counts.

4.2 A small overall difference can contain a larger group loss

For ResNet-18/Waterbirds, the overall gap of -0.098 ± 0.021 pp includes a -0.883 ± 0.784-pp gap for waterbirds on land. The group gap is negative for each original-model mean. This group contains 642 of 5,794 test images, giving it weight 0.111 in Eq. 2. Weighting by this test proportion reduces the contribution of its group gap to the overall difference.

Group accuracy is 71.60% under O, 58.90% under C, and 59.78% under D. The original-model reference exposes deterioration shared by both paths, while $C - D$ measures their additional difference. Absolute group accuracy keeps the level of performance visible in both comparisons.

4.3 The same group can gain or lose across model configurations

The group differences in Figure 2 include positive and negative values. CelebA group 3, blond-haired examples with Male equal to one, has $C - D$ differences of -0.370 ± 0.668, $+0.864 \pm 0.283$, and -0.432 ± 0.283 pp for ResNet-18, DenseNet-121, and ConvNeXt-Tiny. The group contains 180 test examples, so one changed prediction corresponds to 0.556 pp in a single run. Its response depends on the evaluated model and pruning recipe.

4.4 From group outcomes to individual predictions

Within a group, the two endpoints can achieve close accuracy while being correct on different examples. For waterbirds on land, correctness disagreement between C and D is $7.25 \pm 0.74\%$: exactly

one endpoint is correct on these examples, despite the 0.883-pp accuracy difference. Accuracy difference subtracts the rates for only C correct and only D correct; correctness disagreement adds them. RQ3 asks which corrections and losses produce this balance, using O as the reference and P to identify pruning damage.

5 Individual predictions: corrections, negative flips, and repair

5.1 Corrections and negative flips relative to the original model

To resolve the group differences, let $o_i, p_i, c_i, d_i \in \{0, 1\}$ indicate correctness on test example i under O, P, C, D, with 1 meaning correct; m_i denotes correctness under a model M.

Relative to O, a correction makes an original error correct, and a negative flip makes an originally correct prediction incorrect. Following Yan et al. (2021), their rates for a final model M are

$$\mathrm{NFR}_O(M) = \widehat{\Pr}(o_i = 1, m_i = 0), \qquad \mathrm{Corr}_O(M) = \widehat{\Pr}(o_i = 0, m_i = 1), \tag{3}$$

where $\widehat{\Pr}$ is the empirical fraction of all N test examples and NFR denotes negative-flip rate. Their difference gives the accuracy change:

$$\begin{aligned} \mathrm{Acc}(M) - \mathrm{Acc}(O) &= \mathrm{Corr}_O(M) - \mathrm{NFR}_O(M), \\ \mathrm{Acc}(C) - \mathrm{Acc}(D) &= \underbrace{\mathrm{Corr}_O(C) - \mathrm{Corr}_O(D)}_{G\text{ : extra corrections}} - \underbrace{\big[\mathrm{NFR}_O(C) - \mathrm{NFR}_O(D)\big]}_{L\text{ : extra negative flips}}. \end{aligned} \tag{4}$$

Positive G and L denote more corrections and negative flips than continuation; both differences are signed. Rates use percent and differences pp. Group rates use N_g, giving $\mathrm{Acc}_g(C) - \mathrm{Acc}_g(D) = G_g - L_g$ and connecting individual changes to Eq. 2.

5.2 Extra corrections and extra negative flips offset each other

Eight of nine primary setting means have both $G > 0$ and $L > 0$ (Figure 3). All nine have positive mean G; ConvNeXt-Tiny/CelebA has negative mean L. Both signs are positive in 20 of 27 original models and in all three original models of three settings. In these settings, the small net accuracy differences therefore accompany more corrections and more losses relative to O.

For ResNet-18/Waterbirds, $G = 0.345 \pm 0.097$ pp and $L = 0.443 \pm 0.100$ pp give the -0.098-pp net difference. Within the waterbirds-on-land group, $G = 0.346 \pm 0.197$ pp and $L = 1.229 \pm 0.975$ pp yield the -0.883-pp difference (Figure 3b).

All four CIFAR-100 condition means also have $G, L > 0$, with cross-entropy used throughout training. With random 20% pruning, $G = 0.863 \pm 0.171$ pp and $L = 1.382 \pm 0.208$ pp (Table 26).

Endpoint and data choices also change this pattern (Table 10). With original training data, five settings have both $G, L > 0$ after validation selection. With the repair subset, six have both signs positive under each endpoint rule. These components explain the accuracy comparisons in Section 3.4.

As a reference for training randomness, we measure label disagreement, the fraction of examples receiving different predicted labels across training seeds (Bahri & Jiang, 2021). We compare recovered and continuation models (between-path disagreement) and models from the same path (within-path disagreement). In both data regimes, mean between-path disagreement exceeds average within-path disagreement for every main-grid setting (Appendix D).

5.3 Separating repair from other prediction changes

To explain recovery gain, we identify damage repair using the immediately pruned model P. The damage cohort $\mathcal{A} = \{i : o_i = 1, p_i = 0\}$ contains examples made incorrect by pruning. The initially spared cohort $\mathcal{S} = \{i : o_i = 1, p_i = 1\}$ contains examples correct before and after pruning. Membership is fixed before recovery.

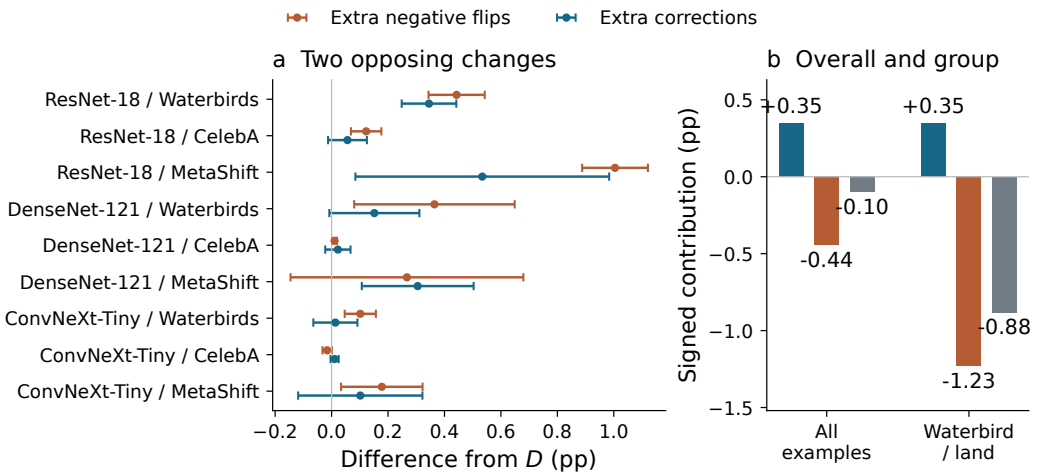


Figure 3: **Net accuracy differences combine extra corrections and extra negative flips.** (a) Extra negative flips L and extra corrections G relative to matched continuation, for all original-data, fixed-budget settings. Points and bars are mean ± original-model SD ($n = 3$, each averaging three repeats). (b) ResNet-18/Waterbirds: blue is G, orange is $-L$, and gray is $G - L$, for all examples and for waterbirds on land. Each component uses the full displayed population as denominator.

The persistent error cohort $\mathcal{E} = \{i : o_i = 0, p_i = 0\}$ contains original errors that remain incorrect after pruning. For nonempty $\mathcal{A}$, the damage-conditioned repair rate under model M is

$$r_{\mathcal{A}}(M) = \frac{\#\{i : o_i = 1, p_i = 0, m_i = 1\}}{\#\mathcal{A}}. \tag{5}$$

This fraction measures damage repair under C and correctness on the same examples under D. The cohort size $\#\mathcal{A}$ and test-set share specify the number of predictions represented. Aggregate repair weights group repair rates by their damaged-example counts; group accuracy uses group sizes.

For ResNet-18/Waterbirds, damage repair adds 2.113 pp and corrections in $\mathcal{E}$ add 1.147 pp. Losses in $\mathcal{S}$ subtract 2.081 pp, and losses on predictions made correct by pruning subtract 0.779 pp. Their net contribution is the 0.40-pp recovery gain, combining repair with corrections and losses elsewhere. Appendix C.2 reports event definitions and variation across original models.

5.4 The continuation reference changes where losses are concentrated

We compare cohort contributions to negative flips under C with their additional contributions relative to D. Unresolved pruning damage at the endpoint consists of examples in $\mathcal{A}$ that are incorrect under the final model. Together with losses in $\mathcal{S}$, it accounts for negative flips.

$$\mathrm{NFR}_O(M) = \underbrace{\widehat{\Pr}(o_i = 1, p_i = 0, m_i = 0)}_{\text{unresolved pruning damage}} + \underbrace{\widehat{\Pr}(o_i = 1, p_i = 1, m_i = 0)}_{\text{loss in the initially spared cohort}}. \tag{6}$$

Both terms use the whole-test denominator N. The first equals $(\#\mathcal{A}/N)[1 - r_{\mathcal{A}}(M)]$, linking damage repair to overall accuracy. For D, the partition evaluates continuation on the same cohorts.

The initially spared cohort contributes more negative flips in eight of the nine primary setting means (Table 18). After subtracting continuation losses, the damage cohort contributes more extra negative flips in six (Table 19). The ordering switches in five settings.

For ResNet-18/Waterbirds, the cohort contributions are (mean ± SD):

Cohort	Negative flips under C (%)	Extra negative flips $C - D$ (pp)
Damage $\mathcal{A}$	1.408 ± 0.042	0.393 ± 0.034
Initially spared $\mathcal{S}$	2.081 ± 0.347	0.050 ± 0.067

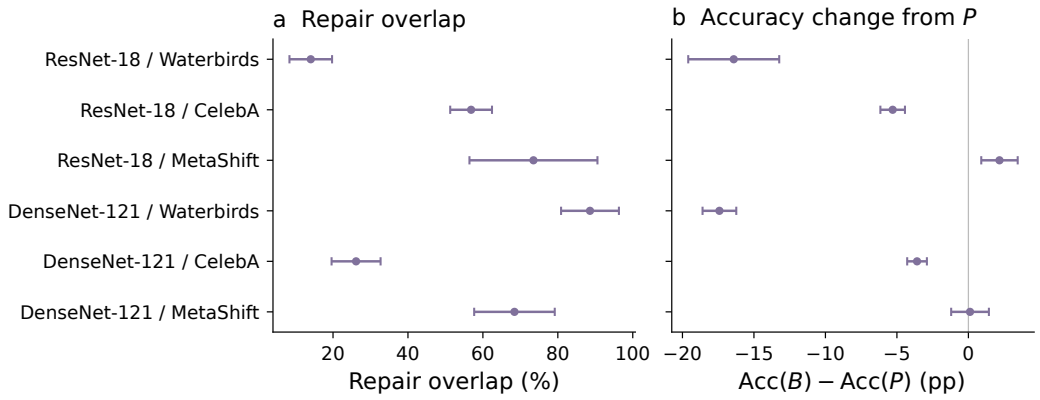


Figure 4: **High repair overlap can accompany lower overall accuracy.** Model B updates only BatchNorm running means and variances in P, with learned parameters fixed. (a) Repair overlap: the percentage of predictions repaired by C in the damage cohort that B also predicts correctly. (b) Overall accuracy change $B - P$. All six original-data settings; mean $\pm$ SD across three original models. Ratios are averaged within original models; B is shared across recovery repeats.

Both columns use the whole-test denominator. Similar losses under D in $\mathcal{S}$ make $\mathcal{A}$ the larger contributor to extra negative flips.

The small extra negative-flip rate in $\mathcal{S}$ also combines different predictions. Examples wrong only under C account for 0.882% of the test set, and those wrong only under D for 0.832%, giving 0.050 pp. Another 1.199% are wrong under both endpoints. Common errors cancel; the difference comes from the two sets of unique errors (Table 20).

5.5 Repair within the damage cohort can accompany losses elsewhere

We examine how damage repair relates to overall accuracy using a BatchNorm statistics-only update (Ioffe & Szegedy, 2015; Li et al., 2020). Models B and C start independently from P. Model B re-estimates BatchNorm running statistics over 32 augmented batches of 32 recovery-data examples, with all learned parameters fixed. The control applies to ResNet-18 and DenseNet-121. Appendix C.5 specifies resets and averaging.

For DenseNet-121/Waterbirds, the damage-conditioned repair rate is $88.03 \pm 8.73\%$ for B and $70.71 \pm 5.86\%$ for C. Repair overlap measures the fraction of damage-cohort predictions repaired by C that B also predicts correctly: $88.59 \pm 7.72\%$ in this setting (Figure 4a). At the same time, $\mathrm{Acc}(B) - \mathrm{Acc}(P) = -17.41 \pm 1.18$ pp, and $22.66 \pm 1.88\%$ of all test examples are correct under P and wrong under B. Here, high damage repair coexists with losses outside the damage cohort that reduce overall accuracy.

Tables 21–23 report cohort sizes, repair rates, and full-test-set outcomes. Statistics use uniform sampling with replacement from imbalanced recovery data.

5.6 Retention of repaired predictions

Figure 5 tracks damage-cohort correctness during CIFAR-100 training. Final correctness under magnitude 10% pruning is $83.76 \pm 1.50\%$ for recovery and $88.54 \pm 0.72\%$ for continuation. Continuation has higher final correctness in all four condition means.

A renewed error occurs when a prediction is correct initially or at an earlier saved checkpoint and is incorrect at the current checkpoint. For recovery on the damage cohort, this records repair followed by another error (Toneva et al., 2019). Under magnitude 10% pruning, these predictions account for $0.312 \pm 0.057\%$ of the whole test set at the final checkpoint (14,080 updates). Both paths exhibit renewed errors, including in the persistent error cohort (Appendix E.1).

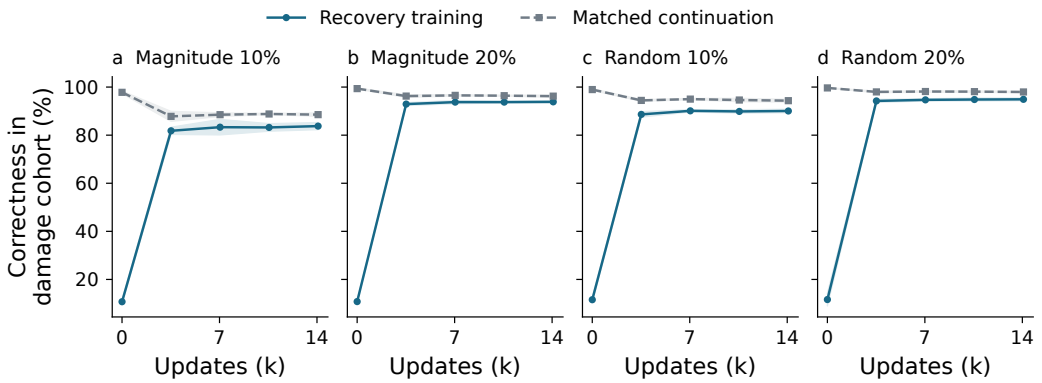


Figure 5: **Repair and continuation on the same damaged predictions.** CIFAR-100/ResNet-18, all four pruning conditions. Panels track current correctness within their fixed $\mathcal{A} = \{o_i = 1, p_i = 0\}$ cohort. At update zero, C is wrong and D correct throughout this cohort; points start after the first update. Lines and bands are mean $\pm$ SD over three original models, each averaging two repeats. Each condition has its own cohort, shared by both paths.

5.7 Implications for evaluating recovered models

A small L can combine losses on different examples (Section 5.4). We define a continuation-referenced retention diagnostic.

$$R_{\text{keep}}(C; O, D) = \frac{\sum_i o_i d_i c_i}{\sum_i o_i d_i}. \tag{7}$$

This rate measures correctness under C on examples correct under both O and D. Report the reference-set size and treat empty sets as undefined. As one minus a conditional negative-flip rate (Yan et al., 2021), it measures endpoint retention. Overall and group accuracy and negative flips relative to O expose shared deterioration. Section 5.6 examines retention during training.

At fixed computation, we propose maximizing R_{keep} among checkpoints meeting prespecified overall and group-accuracy requirements. Keep O, D fixed across candidates. Across pruning levels, identify the largest tested FLOP saving meeting these requirements and a retention threshold. Appendix F gives the loss decomposition, constrained criterion, and validation protocol.

6 Conclusion

This study investigates how recovery training repairs pruning damage and retains correct predictions in compressed models. Analyses of overall performance, group outcomes, and individual prediction histories show that recovery gains combine damage repair, corrections of original errors, and losses of correct predictions. These changes can offset in overall accuracy while leaving larger differences within particular groups. High repair within the damage cohort can also accompany lower overall accuracy when losses elsewhere outweigh the gains. Comparing losses with matched continuation can change the ordering of cohort loss contributions, while the original-model reference reveals deterioration shared by both training paths. Renewed errors after repair further demonstrate the importance of preserving repaired predictions. Together, these findings explain recovery quality through the composition of gains, the distribution of losses, and the persistence of repairs. Based on these findings, we define a continuation-referenced retention diagnostic that makes prediction retention explicit alongside overall and group accuracy and negative flips relative to the original model. This joint assessment provides criteria for evaluating recovery training and selecting compressed models according to computational savings, predictive performance, and prediction retention.

Future work focuses on evaluating the diagnostic for recovery-checkpoint selection against criteria based on overall accuracy, worst-group accuracy, and negative flips. Extending this evaluation across pruning levels and recovery strategies can further establish its value for selecting compressed models.

AI USE STATEMENT

Generative AI tools were used for manuscript editing, reference verification, and supporting code revisions. The authors take responsibility for the final content of the paper.

REFERENCES

Dara Bahri and Heinrich Jiang. Locally Adaptive Label Smoothing Improves Predictive Churn. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 532–542, 2021. URL https://proceedings.mlr.press/v139/bahri21a.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://www.jmlr.org/papers/v13/gretton12a.html.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1135–1143, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/cvpr.2016.90.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What Do Compressed Deep Neural Networks Forget? *arXiv preprint arXiv:1911.05248*, 2019. doi: 10.48550/arXiv.1911.05248. URL https://arxiv.org/abs/1911.05248.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, 2017. doi: 10.1109/cvpr.2017.243.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 448–456, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Tian Jin, Michael Carbin, Daniel M. Roy, Jonathan Frankle, and Gintare Karolina Dziugaite. Pruning's Effect on Generalization Through the Lens of Training and Regularization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 37947–37961, 2022. doi: 10.52202/068431-2750.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2204.02937.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. URL https://cave.cs.toronto.edu/kriz/cifar.html.

Bailin Li, Bowen Wu, Jiang Su, and Guangrun Wang. EagleEye: Fast Sub-net Evaluation for Efficient Neural Network Pruning. In *Computer Vision – ECCV 2020*, volume 12347 of *Lecture Notes in Computer Science*, pp. 639–654. Springer, Cham, 2020. doi: 10.1007/978-3-030-58536-5_38.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning Filters for Efficient ConvNets. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1608.08710.

Weixin Liang and James Zou. MetaShift: A Dataset of Datasets for Evaluating Contextual Distribution Shifts and Training Conflicts. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.06523.

Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. Lost in Pruning: The Effects of Pruning Neural Networks beyond Test Accuracy. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 93–138, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/521437c574a2bb7fcc20b222700b4181-Abstract.html.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning Efficient Convolutional Networks through Network Slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2755–2763, 2017. doi: 10.1109/iccv.2017.298.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the Value of Network Pruning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.05270.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11966–11976, 2022. doi: 10.1109/cvpr52688.2022.01167.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015. doi: 10.1109/iccv.2015.425.

Pavlo Molchanov, Stephen Tyree, Timo Karras, Timo Aila, and Jan Kautz. Pruning Convolutional Neural Networks for Resource Efficient Inference. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.06440.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing Rewinding and Fine-tuning in Neural Network Pruning. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2003.02389.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1911.08731.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020. doi: 10.1145/3381831.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Cuong Tran, Ferdinando Fioretto, Jung-Eun Kim, and Rakshit Naidu. Pruning has a disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17652–17664, 2022. doi: 10.52202/068431-1283.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-Congruent Training: Towards Regression-Free Model Updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14294–14303, 2021. doi: 10.1109/cvpr46437.2021.01407.

A DATA, TRAINING, AND STRUCTURED PRUNING

A.1 DATASETS, GROUPS, AND SPLITS

Waterbirds uses the waterbird/landbird label and water/land background. CelebA uses blond hair as the target (the dataset field `Blond_Hair`) and the supplied Male attribute for grouping. MetaShift uses cat/dog labels and indoor/outdoor context. For the group code $g = 2y + a$, $y = 1$ means waterbird, blond hair, or cat, respectively; $a = 1$ means water background, Male, or indoor context. Group membership follows the supplied dataset annotations. The main tasks have two classes and four groups. Classification uses the raw-logit argmax throughout.

Waterbirds and CelebA retain their official training and test sets. We partition each validation set by group-stratified sampling with seed 2038 into disjoint structure-calibration, repair, model-selection, and analysis subsets, in proportions 20%/20%/20%/40% (Table 2). The structure-calibration subset supplies channel scores; the model-selection subset selects checkpoints. Recovery in the original-data regime draws from the original training set; recovery in the repair-subset regime draws from the repair subset. The analysis subset is reserved for evaluation. Original-model training, recovery, and continuation share a frozen split within each original model. The model-selection and analysis subsets correspond to `race_discovery` and `race_adjudication` in the code. The analysis data include hashes identifying the splits and paired sample draws.

MetaShift uses the cat/dog contextual image pool distributed through SubpopBench[1]. Filenames are sorted within directories before a random split with seed 42 assigns approximately 65%/10%/25% to training/validation/test. We partition its validation set by group-stratified sampling with seed 2038 and the same 20%/20%/20%/40% proportions. The pool contains 3,499 unique images. This random split evaluates the four predefined groups within the same contextual pool.

Table 2: Main-grid data sizes. Structure, Repair, Selection, and Analysis partition the validation pool. All six subsets are disjoint.

Dataset	Train	Structure	Repair	Selection	Analysis	Test
Waterbirds	4,795	240	240	240	479	5,794
CelebA	162,770	3,973	3,973	3,973	7,948	19,962
MetaShift	2,276	70	70	70	139	874

The binary models use 224×224 images with the image-processing convention of deep feature reweighting (DFR) (Kirichenko et al., 2023): random crop and horizontal flip for training, deterministic center-crop evaluation, and ImageNet normalization. Recovery and continuation receive the same augmentation seed for each drawn example. Test evaluation uses a fixed view of each example.

CIFAR-100 uses the official 50,000 training and 10,000 test examples. A stratified split with seed 1002026 reserves 50 training images per fine class for validation, leaving 45,000 training images. The test set has 100 images per fine class. Training applies a random 32×32 crop after four-pixel padding and random horizontal flipping; evaluation is deterministic. Normalization uses means $(0.5071, 0.4867, 0.4408)$ and standard deviations $(0.2675, 0.2565, 0.2761)$.

A.2 ORIGINAL MODELS AND RECOVERY TRAINING

Main-grid original models are ImageNet-pretrained ResNet-18, DenseNet-121, and ConvNeXt-Tiny, trained with the official GroupDRO training core under a common local data interface. Original training uses SGD, initial learning rate 0.001, momentum 0.9, weight decay 10^{-4}, batch size 32, and group-weight step size 0.01. It permits at most 50 epochs, with validation after each epoch and model selection beginning at epoch 15. Training stops after eight consecutive eligible epochs without improvement under the selection rule. The cosine learning-rate floor is 1% of the initial rate. Selection prioritizes validation worst-group accuracy. Original-model identifiers are 3, 4, and 5 for

[1]Source repository: `https://github.com/YyzHarry/SubpopBench`.

ResNet-18, 2, 3, and 4 for DenseNet-121, and 11, 12, and 13 for ConvNeXt-Tiny. Each identifier indexes an independently trained original model within its architecture.

Table 3: Torchvision weight identifiers used to initialize the main-grid original models.

Architecture	Weight identifier
ResNet-18	`ResNet18_Weights.IMAGENET1K_V1`
DenseNet-121	`DenseNet121_Weights.IMAGENET1K_V1`
ConvNeXt-Tiny	`ConvNeXt_Tiny_Weights.IMAGENET1K_V1`

Both C and D optimize cross-entropy from their respective starting states. Their matched comparison holds the GroupDRO-to-cross-entropy change fixed between paths. Changes relative to O include the response to this objective change and further data exposure; $C - D$ also includes interactions between the objective and the pruned structure. The CIFAR-100 experiment uses cross-entropy throughout.

Recovery training samples uniformly with replacement. Binary tasks use batch size 32, initial learning rate 10^{-3}, and successful-update budgets of 600 for Waterbirds and MetaShift and 2,500 for CelebA. CIFAR-100 uses a ResNet-18 with a 3×3, stride-one stem and no initial max-pooling. Original training runs from scratch for 70,400 updates with batch size 128 and initial learning rate 0.1. Validation accuracy is evaluated every 352 updates to select the original model. Recovery uses 14,080 updates and initial learning rate 0.01. These budgets correspond to approximately 200 and 40 training-set-sized draws with replacement. The original seeds are 100, 101, and 102; each has two recovery repeats. Pruning conditions share the original model and its continuation reference.

For each main-grid original model, recovery repeat $r \in \{0, 1, 2\}$ uses seed $s_{\text{base}} + r$, with the base seeds in Table 4. Each paired C, D run uses the same seed. The original-data and repair-subset regimes use the same base seed for a given original model and dataset. For CIFAR-100, $s_{\text{base}} = 710000 + 100 s_O$, where $s_O \in \{100, 101, 102\}$ is the original-model seed. The resulting base seeds are 720000, 720100, and 720200; the two recovery repeats use $r \in \{0, 1\}$.

Table 4: Main-grid recovery base seeds s_{base}. Original-model identifiers distinguish the three original models within each architecture.

Architecture	Original-model ID	Waterbirds	CelebA	MetaShift
ResNet-18	3	483000	483300	483600
	4	483100	483400	483700
	5	483200	483500	483800
DenseNet-121	2	483900	484200	484500
	3	484000	484300	484600
	4	484100	484400	484700
ConvNeXt-Tiny	11	484800	485100	485400
	12	484900	485200	485500
	13	485000	485300	485600

For successful update $t \in \{1, \ldots, T\}$, the learning rate is

$$\eta_t = \eta_0 \left[0.01 + 0.99 \frac{1 + \cos\{\pi(t-1)/T\}}{2} \right], \tag{8}$$

where T is the update budget and η_0 the initial learning rate. SGD momentum is 0.9, weight decay is 10^{-4}, and gradient-norm clipping uses a maximum norm of 10. Each paired run starts with a fresh optimizer. Example draws and augmentation seeds are matched; stochastic modules inside networks of different shapes can consume random numbers differently.

Training uses automatic mixed precision. Non-finite scaled-gradient attempts are retried with reduced scaling while preserving the batch, random state, and model buffers. Only successful updates advance the budget. Evaluation uses 32-bit floating-point arithmetic (FP32), with TensorFloat-32 (TF32) disabled. The recorded environment is Python 3.11.9, PyTorch 2.6.0 with CUDA 12.6, and an NVIDIA RTX 3090.

A.3 ENDPOINT SELECTION AND SAVED CHECKPOINTS

Original-model and recovery checkpoint selection both use the model-selection subset, accessed through `race_discovery`. Binary recovery validation is evaluated every 50 updates. Selection maximizes worst-group accuracy, then overall accuracy, then prefers the earlier update. CIFAR-100 uses its 5,000-example validation set through the same loader field, with overall accuracy and the earlier update for a tie. Fixed endpoints use the final successful update. Thus, fixed endpoints match realized update counts, and selected endpoints apply the same validation rule within a common maximum budget.

Table 5: Group counts in the frozen model-selection split, in group-code order. The final column is $100/\min_g N_g^{\text{selection}}$, the accuracy increment from one example in the smallest selection group.

Dataset	Group 0	Group 1	Group 2	Group 3	Increment (pp)
Waterbirds	93	93	27	27	3.704
CelebA	1707	1655	575	36	2.778
MetaShift	25	15	7	23	14.286

These counts come from the frozen split manifests. Worst-group accuracy changes in discrete increments, particularly on MetaShift. Differences between fixed and selected endpoints reflect validation variation, the selection rule, and training dynamics.

Table 6: Sample exposure per recovery repeat. Training samples with replacement. Recovery and continuation have the same exposure within each data regime.

Dataset	Drawn positions	Original-set-sized draws	Repair-set-sized draws
Waterbirds	19200	4.004	80.000
CelebA	80000	0.492	20.136
MetaShift	19200	8.436	274.286

The original-data and repair-subset regimes differ in both empirical composition and exposure per example. For CelebA, the original-data budget samples 0.492 times the training-set size, whereas the repair-subset budget samples 20.136 times the subset size.

Prediction snapshots are saved after the first successful update and at 25%, 50%, 75%, and 100% of the budget. Trajectory events describe changes between these saved checkpoints.

A.4 STRUCTURED PRUNING AND COMPUTATIONAL REDUCTION

The main intervention removes channels confined to the interior of each block; we call these private channels because removing them preserves block input/output shapes. ResNet-18 removes channels inside basic residual blocks and the corresponding normalization and downstream-input parameters. DenseNet-121 removes private bottleneck channels, preserving concatenation interfaces. ConvNeXt-Tiny removes expanded hidden channels within its block multilayer perceptrons (MLPs). Each intervention produces an executable network with reduced internal width.

The group-sensitive channel score is based on the squared first-order Taylor estimate of the change in group mean loss caused by channel removal. Following the activation-gradient construction of Taylor pruning (Molchanov et al., 2017), we compute this estimate from mean cross-entropy within each calibration group. For layer ℓ, channel c, group g, and batch b, let $a_{i\ell cs}$ denote an activation at spatial position s and let $\mathcal{L}_{gb}$ be mean loss on that batch's examples in group g. Define

$$q_{\ell cgb} = \left(\sum_{i,s} a_{i\ell cs} \frac{\partial \mathcal{L}_{gb}}{\partial a_{i\ell cs}} \right)^2, \qquad v_{\ell c} = \max_g \frac{1}{|\mathcal{B}_g|} \sum_{b \in \mathcal{B}_g} q_{\ell cgb}, \tag{9}$$

where $\mathcal{B}_g$ comprises processed batches containing group g. For n_{gb} examples in the group and per-example loss ℓ_i, write $u_{i\ell c} = \sum_s a_{i\ell cs}\, \partial \ell_i / \partial a_{i\ell cs}$. Then $q_{\ell cgb} = (n_{gb}^{-1} \sum_{i \in (g,b)} u_{i\ell c})^2$: the mean is

taken before squaring. Averaging these squared values over batches and taking the maximum over groups gives $v_{\ell c}$; the architecture-specific rules below convert it to the group-sensitive channel score $s_{\ell c}$.

For ResNet-18 and DenseNet-121, activations are taken at private-channel normalization outputs and the score is $s_{\ell c} = \sqrt{v_{\ell c}}$. At most 16 structure-calibration batches are used. Each layer orders channels by ascending score and index. Each candidate removes the first k channels in this ordering, a prefix. Allowed k are multiples of eight that leave at least 16 channels. A multiple-choice dynamic program jointly selects one prefix per layer. Its lexicographic objective first minimizes absolute deviation from the FLOP target, then total removed-channel score, then the ordered channel-count tuple. FLOP costs are discretized by their common divisor.

For ConvNeXt-Tiny, activations are taken after the expansion linear layer. Sixteen seeded, group-balanced calibration batches contain two examples per group. The score is $s_{\ell c} = v_{\ell c} / \max(\mathrm{mean}_{c'}\, v_{\ell c'}, 10^{-12})$. Channels are ranked globally by this normalized score, then by decreasing per-channel FLOP savings, module name, and index. Removal stops at the target, with at most 50% of each layer's original expanded hidden channels eligible for removal. These architecture-specific scoring and allocation rules define the pruning recipes used in the comparison.

Structure construction uses the structure-calibration subset. The resulting structure is fixed across recovery repeats and both recovery-data regimes for each original model. CIFAR-100 uses magnitude and fixed-seed random channel orderings. Magnitude combines the squared outgoing weights, corresponding downstream-input weights, and BatchNorm affine parameters before a square root. Candidate channel sets are prefixes of these orderings, and the allocation program selects layer widths for the 10% or 20% FLOP target. The random ordering uses seed $610000 + s_O$, where $s_O \in \{100, 101, 102\}$ is the original-model seed, giving seeds 610100, 610101, and 610102. Each original model uses the same ordering for both FLOP targets and across both recovery repeats.

Table 7: Measured main-grid structural reduction. Ranges span all three original models. FLOPs and parameters each use their own original-model value as denominator.

Architecture / dataset	Counted FLOPs removed (%)	Parameters removed (%)
ResNet-18 / Waterbirds	9.993–9.995	7.918–12.124
ResNet-18 / CelebA	9.995–9.995	7.176–8.907
ResNet-18 / MetaShift	9.993–9.995	7.918–8.908
DenseNet-121 / Waterbirds	10.045–10.045	2.979–3.233
DenseNet-121 / CelebA	10.045–10.047	2.393–3.267
DenseNet-121 / MetaShift	10.043–10.044	3.422–5.092
ConvNeXt-Tiny / Waterbirds	10.002–10.002	26.780–26.927
ConvNeXt-Tiny / CelebA	10.001–10.009	22.135–24.014
ConvNeXt-Tiny / MetaShift	10.001–10.006	26.378–26.691

Operation counts use `fvcore==0.1.5.post20221221`, with the same convention and input shape before and after pruning. Unsupported pooling and elementwise-addition operators are omitted consistently. Parameter reduction differs across architectures because removing channels at different resolutions and block positions changes arithmetic and storage differently.

B Complete endpoint and group results

Table 8 reports every architecture and dataset under both recovery-data regimes and endpoint rules. Table 9 resolves the same comparisons into extra negative flips L and extra corrections G.

Table 8: $C - D$ test-accuracy differences in pp. Mean $\pm$ SD across three original models, after averaging three recovery repeats. Selected endpoints follow the validation worst-group accuracy rule in Appendix A.3.

Architecture / dataset	Original: fixed	Original: selected	Subset: fixed	Subset: selected
ResNet-18 / Waterbirds	-0.098 ± 0.021	-1.220 ± 0.558	-0.313 ± 0.235	-0.374 ± 0.527
ResNet-18 / CelebA	-0.066 ± 0.053	-0.178 ± 0.149	-0.035 ± 0.029	-0.053 ± 0.462
ResNet-18 / MetaShift	-0.470 ± 0.333	-0.877 ± 0.303	-1.322 ± 1.245	-1.754 ± 0.250
DenseNet-121 / Waterbirds	-0.213 ± 0.371	-0.591 ± 0.283	0.008 ± 0.144	-0.029 ± 0.015
DenseNet-121 / CelebA	0.012 ± 0.042	0.054 ± 0.067	-0.011 ± 0.018	-0.004 ± 0.135
DenseNet-121 / MetaShift	0.038 ± 0.413	-0.178 ± 0.461	-0.064 ± 0.210	-0.140 ± 0.425
ConvNeXt-Tiny / Waterbirds	-0.088 ± 0.072	-0.368 ± 0.571	-0.058 ± 0.061	-0.113 ± 0.201
ConvNeXt-Tiny / CelebA	0.026 ± 0.022	-0.040 ± 0.075	-0.036 ± 0.056	0.134 ± 0.092
ConvNeXt-Tiny / MetaShift	-0.076 ± 0.212	-0.331 ± 0.397	-0.076 ± 0.815	-0.254 ± 0.354

The joint sign count in Table 9 records how many of the three original models have both $G > 0$ and $L > 0$. Each endpoint and data choice reuses these original models.

Table 9: Extra corrections and extra negative flips across data and endpoint choices. All differences are in pp; mean $\pm$ SD across three original models, each averaging three recovery repeats. Joint sign count is the number of original models with both $L > 0$ and $G > 0$.

Architecture	Dataset	L	G	$C - D$	Joint $+/3$
Original / fixed					
ConvNeXt-Tiny	CelebA	-0.016 ± 0.017	0.011 ± 0.014	0.026 ± 0.022	1
ConvNeXt-Tiny	MetaShift	0.178 ± 0.144	0.102 ± 0.220	-0.076 ± 0.212	2
ConvNeXt-Tiny	Waterbirds	0.102 ± 0.055	0.013 ± 0.078	-0.088 ± 0.072	2
DenseNet-121	CelebA	0.011 ± 0.009	0.022 ± 0.045	0.012 ± 0.042	2
DenseNet-121	MetaShift	0.267 ± 0.413	0.305 ± 0.198	0.038 ± 0.413	2
DenseNet-121	Waterbirds	0.364 ± 0.284	0.151 ± 0.160	-0.213 ± 0.371	3
ResNet-18	CelebA	0.122 ± 0.054	0.056 ± 0.069	-0.066 ± 0.053	2
ResNet-18	MetaShift	1.004 ± 0.117	0.534 ± 0.450	-0.470 ± 0.333	3
ResNet-18	Waterbirds	0.443 ± 0.100	0.345 ± 0.097	-0.098 ± 0.021	3
Original / selected					
ConvNeXt-Tiny	CelebA	-0.150 ± 0.108	-0.189 ± 0.182	-0.040 ± 0.075	0
ConvNeXt-Tiny	MetaShift	0.292 ± 0.418	-0.038 ± 0.232	-0.331 ± 0.397	1
ConvNeXt-Tiny	Waterbirds	0.290 ± 0.409	-0.079 ± 0.162	-0.368 ± 0.571	1
DenseNet-121	CelebA	0.030 ± 0.197	0.083 ± 0.222	0.054 ± 0.067	2
DenseNet-121	MetaShift	0.674 ± 0.320	0.496 ± 0.153	-0.178 ± 0.461	3
DenseNet-121	Waterbirds	0.748 ± 0.345	0.157 ± 0.101	-0.591 ± 0.283	3
ResNet-18	CelebA	0.213 ± 0.046	0.035 ± 0.113	-0.178 ± 0.149	2
ResNet-18	MetaShift	1.691 ± 0.022	0.814 ± 0.318	-0.877 ± 0.303	3
ResNet-18	Waterbirds	1.158 ± 0.291	-0.061 ± 0.276	-1.220 ± 0.558	1
Subset / fixed					
ConvNeXt-Tiny	CelebA	0.023 ± 0.011	-0.012 ± 0.045	-0.036 ± 0.056	1
ConvNeXt-Tiny	MetaShift	0.127 ± 0.606	0.051 ± 0.254	-0.076 ± 0.815	1
ConvNeXt-Tiny	Waterbirds	0.025 ± 0.037	-0.033 ± 0.027	-0.058 ± 0.061	0
DenseNet-121	CelebA	0.034 ± 0.043	0.023 ± 0.036	-0.011 ± 0.018	2
DenseNet-121	MetaShift	0.420 ± 0.212	0.356 ± 0.260	-0.064 ± 0.210	3
DenseNet-121	Waterbirds	-0.006 ± 0.134	0.002 ± 0.078	0.008 ± 0.144	1
ResNet-18	CelebA	0.079 ± 0.032	0.044 ± 0.033	-0.035 ± 0.029	3
ResNet-18	MetaShift	1.665 ± 0.844	0.343 ± 0.712	-1.322 ± 1.245	2
ResNet-18	Waterbirds	0.316 ± 0.260	0.004 ± 0.148	-0.313 ± 0.235	2
Subset / selected					
ConvNeXt-Tiny	CelebA	-0.061 ± 0.036	0.073 ± 0.061	0.134 ± 0.092	0
ConvNeXt-Tiny	MetaShift	0.407 ± 0.425	0.153 ± 0.132	-0.254 ± 0.354	2
ConvNeXt-Tiny	Waterbirds	0.090 ± 0.127	-0.023 ± 0.087	-0.113 ± 0.201	0
DenseNet-121	CelebA	0.011 ± 0.144	0.007 ± 0.039	-0.004 ± 0.135	1
DenseNet-121	MetaShift	0.572 ± 0.153	0.432 ± 0.425	-0.140 ± 0.425	3
DenseNet-121	Waterbirds	0.036 ± 0.079	0.008 ± 0.066	-0.029 ± 0.015	2
ResNet-18	CelebA	0.158 ± 0.288	0.105 ± 0.175	-0.053 ± 0.462	1
ResNet-18	MetaShift	1.920 ± 0.318	0.165 ± 0.324	-1.754 ± 0.250	2
ResNet-18	Waterbirds	0.255 ± 0.473	-0.119 ± 0.373	-0.374 ± 0.527	1

Table 10: **Extra corrections and extra negative flips across data and endpoint choices.** Setting counts with mean $G, L > 0$. All regimes use the same nine settings, each with three original models and three repeats.

Recovery data	Fixed endpoint	Validation-selected
Original training set	8/9	5/9
Repair subset	6/9	6/9

B.1 PREDEFINED GROUPS

Table 11 gives the group counts and accuracies of the original model O, recovered model C, and continuation model D for all groups in the nine original-data, fixed-endpoint settings. These accuracies place the differences in Figure 2 at their observed performance levels. Each rate uses the group size as denominator. We average recovery repeats within each original model, then report mean and SD across three original models; $C - D$ is computed within paired runs before aggregation.

Table 11: Accuracy of the original model O, recovered model C, and continuation model D for every predefined group. Original training data and fixed endpoints; N_g is the test count. Accuracy is in percent and $C - D$ in pp. Entries are mean $\pm$ SD across three original models, each averaging three recovery repeats. Group code $g = 2y + a$ follows Appendix A.

Architecture	Group	N_g	O (%)	C (%)	D (%)	$C - D$ (pp)
Waterbirds						
ResNet-18	0	2,255	98.76 ± 0.12	99.35 ± 0.06	99.38 ± 0.05	-0.030 ± 0.103
	1	2,255	80.93 ± 0.90	80.16 ± 1.69	80.03 ± 1.44	0.123 ± 0.355
	2	642	71.60 ± 1.68	58.90 ± 1.57	59.78 ± 0.79	-0.883 ± 0.784
	3	642	93.72 ± 0.63	92.38 ± 0.68	92.71 ± 0.48	-0.329 ± 0.448
DenseNet-121	0	2,255	99.19 ± 0.27	99.52 ± 0.06	99.53 ± 0.03	-0.015 ± 0.064
	1	2,255	86.73 ± 3.35	85.55 ± 1.84	85.83 ± 2.05	-0.276 ± 0.799
	2	642	76.74 ± 3.05	69.97 ± 1.23	70.70 ± 2.18	-0.727 ± 1.090
	3	642	95.53 ± 1.18	95.17 ± 0.67	95.34 ± 0.85	-0.173 ± 0.210
ConvNeXt-Tiny	0	2,255	98.86 ± 0.91	99.72 ± 0.01	99.73 ± 0.02	-0.010 ± 0.017
	1	2,255	86.99 ± 1.27	87.11 ± 0.76	87.35 ± 0.83	-0.241 ± 0.378
	2	642	82.87 ± 4.15	73.21 ± 0.52	73.00 ± 0.55	0.208 ± 0.824
	3	642	95.48 ± 0.56	95.21 ± 0.23	95.33 ± 0.32	-0.121 ± 0.167
CelebA						
ResNet-18	0	9,767	95.95 ± 0.33	96.36 ± 0.17	96.48 ± 0.17	-0.125 ± 0.086
	1	7,535	97.43 ± 0.47	99.18 ± 0.02	99.22 ± 0.05	-0.044 ± 0.062
	2	2,480	86.08 ± 1.27	84.13 ± 0.56	84.01 ± 0.72	0.121 ± 0.447
	3	180	65.56 ± 1.92	44.63 ± 4.73	45.00 ± 4.14	-0.370 ± 0.668
DenseNet-121	0	9,767	94.97 ± 0.41	96.34 ± 0.16	96.37 ± 0.12	-0.039 ± 0.101
	1	7,535	97.17 ± 0.78	99.32 ± 0.06	99.35 ± 0.05	-0.031 ± 0.019
	2	2,480	88.97 ± 1.62	85.13 ± 1.13	84.85 ± 1.22	0.278 ± 0.104
	3	180	65.93 ± 4.72	44.07 ± 2.28	43.21 ± 2.30	0.864 ± 0.283
ConvNeXt-Tiny	0	9,767	93.70 ± 1.68	96.44 ± 0.07	96.44 ± 0.06	0.005 ± 0.036
	1	7,535	94.86 ± 1.45	99.10 ± 0.05	99.09 ± 0.06	0.012 ± 0.010
	2	2,480	92.14 ± 1.79	86.60 ± 0.10	86.41 ± 0.17	0.188 ± 0.198
	3	180	83.15 ± 3.16	50.68 ± 1.13	51.11 ± 1.30	-0.432 ± 0.283
MetaShift						
ResNet-18	0	273	94.51 ± 0.97	94.18 ± 0.49	94.22 ± 0.86	-0.041 ± 0.392
	1	191	84.99 ± 2.58	79.52 ± 1.56	79.58 ± 0.91	-0.058 ± 1.411
	2	65	78.46 ± 2.66	78.97 ± 0.89	80.85 ± 3.41	-1.880 ± 3.710
	3	345	84.83 ± 3.65	87.47 ± 1.71	88.24 ± 1.64	-0.773 ± 0.443
DenseNet-121	0	273	94.63 ± 0.56	95.97 ± 0.32	95.81 ± 0.07	0.163 ± 0.254
	1	191	87.26 ± 0.80	86.74 ± 0.46	86.10 ± 1.01	0.640 ± 0.961
	2	65	82.56 ± 2.35	81.03 ± 1.36	81.71 ± 1.94	-0.684 ± 0.592
	3	345	89.86 ± 0.29	90.08 ± 0.28	90.34 ± 0.44	-0.258 ± 0.712
ConvNeXt-Tiny	0	273	94.26 ± 0.92	96.87 ± 0.51	96.66 ± 0.51	0.204 ± 0.508
	1	191	90.05 ± 2.28	89.70 ± 0.76	89.99 ± 0.71	-0.291 ± 0.363
	2	65	86.15 ± 3.08	83.08 ± 1.85	83.42 ± 1.07	-0.342 ± 1.068
	3	345	91.11 ± 1.70	93.01 ± 0.70	93.14 ± 1.14	-0.129 ± 0.486

The following tables resolve each group difference into extra corrections and extra negative flips, in pp. Correctness disagreement is the percentage of group examples on which exactly one of C, D is correct. Group counts and aggregation follow Table 11.

Table 12: Waterbirds: all predefined groups. N_g is the test count. Group code: 2×waterbird + water background.

Architecture	Group	N_g	$C - D$	Extra negative flips	Extra corrections	Disagreement
ResNet-18	0	2255	-0.030 ± 0.103	0.059 ± 0.078	0.030 ± 0.026	0.296 ± 0.077
ResNet-18	1	2255	0.123 ± 0.355	0.581 ± 0.538	0.705 ± 0.222	6.095 ± 0.415
ResNet-18	2	642	-0.883 ± 0.784	1.229 ± 0.975	0.346 ± 0.197	7.252 ± 0.737
ResNet-18	3	642	-0.329 ± 0.448	0.519 ± 0.187	0.190 ± 0.286	2.406 ± 0.346
DenseNet-121	0	2255	-0.015 ± 0.064	0.044 ± 0.039	0.030 ± 0.030	0.192 ± 0.039
DenseNet-121	1	2255	-0.276 ± 0.799	0.522 ± 0.550	0.246 ± 0.302	3.942 ± 0.179
DenseNet-121	2	642	-0.727 ± 1.090	1.073 ± 1.097	0.346 ± 0.295	5.088 ± 0.187
DenseNet-121	3	642	-0.173 ± 0.210	0.225 ± 0.261	0.052 ± 0.187	1.038 ± 0.260
ConvNeXt-Tiny	0	2255	-0.010 ± 0.017	0.000 ± 0.015	-0.010 ± 0.031	0.118 ± 0.030
ConvNeXt-Tiny	1	2255	-0.241 ± 0.378	0.296 ± 0.213	0.054 ± 0.200	1.798 ± 0.481
ConvNeXt-Tiny	2	642	0.208 ± 0.824	-0.225 ± 0.697	-0.017 ± 0.131	2.665 ± 0.765
ConvNeXt-Tiny	3	642	-0.121 ± 0.167	0.104 ± 0.137	-0.017 ± 0.060	0.571 ± 0.226

Table 13: CelebA: all predefined groups. Group code: 2×blond hair + Male attribute.

Architecture	Group	N_g	$C - D$	Extra negative flips	Extra corrections	Disagreement
ResNet-18	0	9767	-0.125 ± 0.086	0.159 ± 0.019	0.034 ± 0.068	1.044 ± 0.084
ResNet-18	1	7535	-0.044 ± 0.062	0.029 ± 0.018	-0.015 ± 0.045	0.425 ± 0.039
ResNet-18	2	2480	0.121 ± 0.447	0.228 ± 0.491	0.349 ± 0.190	4.539 ± 0.444
ResNet-18	3	180	-0.370 ± 0.668	0.556 ± 1.156	0.185 ± 0.490	9.012 ± 1.020
DenseNet-121	0	9767	-0.039 ± 0.101	0.059 ± 0.019	0.020 ± 0.083	0.712 ± 0.072
DenseNet-121	1	7535	-0.031 ± 0.019	0.004 ± 0.004	-0.027 ± 0.015	0.211 ± 0.051
DenseNet-121	2	2480	0.278 ± 0.104	-0.112 ± 0.097	0.166 ± 0.021	2.464 ± 0.277
DenseNet-121	3	180	0.864 ± 0.283	-0.679 ± 0.466	0.185 ± 0.185	4.938 ± 0.913
ConvNeXt-Tiny	0	9767	0.005 ± 0.036	0.007 ± 0.012	0.011 ± 0.033	0.319 ± 0.033
ConvNeXt-Tiny	1	7535	0.012 ± 0.010	-0.001 ± 0.003	0.010 ± 0.009	0.159 ± 0.018
ConvNeXt-Tiny	2	2480	0.188 ± 0.198	-0.179 ± 0.196	0.009 ± 0.041	1.165 ± 0.099
ConvNeXt-Tiny	3	180	-0.432 ± 0.283	0.432 ± 0.283	0.000 ± 0.000	2.407 ± 0.490

Table 14: MetaShift random-split variant: all predefined groups. Group code: 2×cat + indoor context.

Architecture	Group	N_g	$C - D$	Extra negative flips	Extra corrections	Disagreement
ResNet-18	0	273	-0.041 ± 0.392	0.651 ± 0.615	0.611 ± 0.323	2.401 ± 0.187
ResNet-18	1	191	-0.058 ± 1.411	1.047 ± 0.629	0.989 ± 0.787	6.341 ± 1.008
ResNet-18	2	65	-1.880 ± 3.710	1.709 ± 1.184	-0.171 ± 2.530	4.615 ± 1.357
ResNet-18	3	345	-0.773 ± 0.443	1.127 ± 0.390	0.354 ± 0.148	3.865 ± 0.729
DenseNet-121	0	273	0.163 ± 0.254	-0.000 ± 0.122	0.163 ± 0.352	0.814 ± 0.187
DenseNet-121	1	191	0.640 ± 0.961	0.407 ± 0.504	1.047 ± 0.605	4.596 ± 0.705
DenseNet-121	2	65	-0.684 ± 0.592	1.026 ± 0.513	0.342 ± 1.068	3.077 ± 1.776
DenseNet-121	3	345	-0.258 ± 0.712	0.258 ± 0.866	0.000 ± 0.193	2.834 ± 0.968
ConvNeXt-Tiny	0	273	0.204 ± 0.508	0.163 ± 0.254	0.366 ± 0.323	0.936 ± 0.282
ConvNeXt-Tiny	1	191	-0.291 ± 0.363	-0.058 ± 0.267	-0.349 ± 0.349	2.618 ± 0.972
ConvNeXt-Tiny	2	65	-0.342 ± 1.068	0.513 ± 0.888	0.171 ± 0.296	2.051 ± 1.849
ConvNeXt-Tiny	3	345	-0.129 ± 0.486	0.258 ± 0.201	0.129 ± 0.549	1.739 ± 0.335

C Prediction events and repair

C.1 Counting identities and denominators

For any final classifier M, correction ($o_i = 0, m_i = 1$) adds to accuracy relative to O, and a negative flip ($o_i = 1, m_i = 0$) subtracts from it. Therefore

$$\mathrm{Acc}(M) - \mathrm{Acc}(O) = \mathrm{Corr}_O(M) - \mathrm{NFR}_O(M). \tag{10}$$

Subtracting the expression for D from that for C gives Eq. 4. Applying the same identity within each fixed group and weighting by N_g/N gives Eq. 2.

Table 15: Complete original-data prediction accounting. Extra negative flips L and extra corrections G are in pp; correctness disagreement is in percent. Mean $\pm$ SD across three original models, each averaging three recovery repeats.

Architecture / dataset	Extra negative flips	Extra corrections	Correctness disagreement
ResNet-18 / Waterbirds	0.443 ± 0.100	0.345 ± 0.097	3.557 ± 0.076
ResNet-18 / CelebA	0.122 ± 0.054	0.056 ± 0.069	1.316 ± 0.104
ResNet-18 / MetaShift	1.004 ± 0.117	0.534 ± 0.450	4.005 ± 0.440
DenseNet-121 / Waterbirds	0.364 ± 0.284	0.151 ± 0.160	2.288 ± 0.049
DenseNet-121 / CelebA	0.011 ± 0.009	0.022 ± 0.045	0.779 ± 0.054
DenseNet-121 / MetaShift	0.267 ± 0.413	0.305 ± 0.198	2.606 ± 0.477
ConvNeXt-Tiny / Waterbirds	0.102 ± 0.055	0.013 ± 0.078	1.105 ± 0.239
ConvNeXt-Tiny / CelebA	-0.016 ± 0.017	0.011 ± 0.014	0.382 ± 0.038
ConvNeXt-Tiny / MetaShift	0.178 ± 0.144	0.102 ± 0.220	1.704 ± 0.390

Correctness disagreement is the sum of the rates for "only C correct" and "only D correct." It equals label disagreement in a binary task. In a multiclass task, label disagreement also includes examples on which both models are wrong and predict different classes. The extra negative-flip rate L is the difference between the two endpoints' negative-flip rates relative to O. Correctness turnover relative to O is $T_O(M) = \mathrm{Corr}_O(M) + \mathrm{NFR}_O(M)$. Define $\Delta T = T_O(C) - T_O(D) = G + L$ and $\Delta\,\mathrm{Acc} = \mathrm{Acc}(C) - \mathrm{Acc}(D) = G - L$. Then $G, L > 0$ exactly when $\Delta T > |\Delta\,\mathrm{Acc}\,|$.

With $G > 0, L \leq 0$, recovery achieves more corrections with at most as many negative flips; $G < 0, L \geq 0$ gives the reverse. When G, L share a sign, both rates change together, with accuracy determined by $G - L$.

C.2 Changes during recovery in the worked example

The four joint events in Table 16 account for every accuracy change from P to C. Their signs depend on whether C corrects an error under P or loses a correct prediction under P. The original state O separates repair of pruning damage from correction of an original error, and distinguishes losses in $\mathcal{S}$ from loss of predictions made correct by pruning. The numerical values use the same frozen event counts and hierarchical aggregation as the main-grid results.

Table 16: Accuracy changes during recovery for ResNet-18/Waterbirds, original training data and fixed endpoints. Each event is divided by the whole-test size and signed by its contribution to $\mathrm{Acc}(C) - \mathrm{Acc}(P)$, in pp. Mean $\pm$ SD across three original models, each averaging three recovery repeats.

Prediction change	Event	Contribution (pp)
Repair in the damage cohort	$o = 1, p = 0, c = 1$	2.113 ± 1.329
Correction in the persistent error cohort	$o = 0, p = 0, c = 1$	1.147 ± 0.387
Loss in the initially spared cohort	$o = 1, p = 1, c = 0$	-2.081 ± 0.347
Loss of predictions made correct by pruning	$o = 0, p = 1, c = 0$	-0.779 ± 0.085
Recovery gain	$\mathrm{Acc}(C) - \mathrm{Acc}(P)$	0.401 ± 2.036

C.3 Damage-conditioned repair and negative flips

For $M \in \{C, D\}$, write $M_A = 100\widehat{\Pr}(o_i = 1, p_i = 0, m_i = 0)$ and $M_S = 100\widehat{\Pr}(o_i = 1, p_i = 1, m_i = 0)$. Each is a percentage of the whole test set. Their sum is the negative-flip rate relative to O, expressed as a percentage. The paired differences $C_A - D_A$ and $C_S - D_S$ sum to L. The same fixed cohorts $\mathcal{A}, \mathcal{S}$ are used for both endpoints.

Tables 17, 18, and 19 report all nine primary settings. Conditional rates are calculated for each recovery repeat, averaged within each original model, then summarized across original models. Every damage cohort is nonempty. ConvNeXt-Tiny/MetaShift has four to nine damaged examples, giving each example a large contribution to the conditional repair rate.

The ordering comparison uses mean C_A and C_S, followed by mean paired differences $C_A - D_A$ and $C_S - D_S$. Their order reverses in five settings. These whole-test rates combine cohort size and

error frequency within the cohort. In particular, $|C_A - D_A| \leq 100\#\mathcal{A}/N$, since each endpoint can lose at most all examples in $\mathcal{A}$.

Table 17: Damage-cohort size and final correctness within the damage cohort. $r_{\mathcal{A}}(C)$ is repair after pruning; $r_{\mathcal{A}}(D)$ is correctness after continuation on the same examples. Rates are percentages, mean $\pm$ SD across three original models, each averaging three recovery repeats.

Architecture / dataset	$\lvert\mathcal{A}\rvert$ range	$100\lvert\mathcal{A}\rvert/N$	$r_{\mathcal{A}}(C)$ (%)	$r_{\mathcal{A}}(D)$ (%)
ResNet-18 / Waterbirds	158–292	3.52 ± 1.32	56.63 ± 13.66	68.85 ± 9.38
ResNet-18 / CelebA	168–249	0.99 ± 0.22	63.00 ± 3.88	69.32 ± 3.59
ResNet-18 / MetaShift	55–70	7.13 ± 0.86	76.20 ± 9.87	86.40 ± 6.51
DenseNet-121 / Waterbirds	152–215	3.28 ± 0.58	70.71 ± 5.86	74.51 ± 8.95
DenseNet-121 / CelebA	134–169	0.74 ± 0.09	73.61 ± 1.49	77.34 ± 1.14
DenseNet-121 / MetaShift	22–66	4.58 ± 2.64	79.69 ± 12.86	85.77 ± 6.51
ConvNeXt-Tiny / Waterbirds	9–13	0.18 ± 0.04	57.45 ± 8.04	65.81 ± 1.48
ConvNeXt-Tiny / CelebA	45–90	0.30 ± 0.13	75.29 ± 13.07	77.19 ± 12.38
ConvNeXt-Tiny / MetaShift	4–9	0.69 ± 0.30	70.00 ± 12.02	71.54 ± 17.49

Table 18: Negative flips by cohort. Every entry is a percentage of the whole test set, mean $\pm$ SD across original models. $C_A + C_S$ and $D_A + D_S$ give the respective negative-flip rates. C_A counts examples in the damage cohort that are incorrect under C.

Architecture / dataset	C_A (%)	C_S (%)	D_A (%)	D_S (%)
ResNet-18 / Waterbirds	1.408 ± 0.042	2.081 ± 0.347	1.014 ± 0.017	2.031 ± 0.287
ResNet-18 / CelebA	0.365 ± 0.077	0.753 ± 0.046	0.302 ± 0.061	0.693 ± 0.031
ResNet-18 / MetaShift	1.653 ± 0.508	1.055 ± 0.134	0.953 ± 0.404	0.750 ± 0.172
DenseNet-121 / Waterbirds	0.968 ± 0.312	1.632 ± 0.881	0.857 ± 0.416	1.379 ± 0.822
DenseNet-121 / CelebA	0.195 ± 0.028	0.806 ± 0.071	0.168 ± 0.028	0.823 ± 0.061
DenseNet-121 / MetaShift	0.814 ± 0.404	1.004 ± 0.425	0.610 ± 0.350	0.941 ± 0.172
ConvNeXt-Tiny / Waterbirds	0.079 ± 0.023	2.274 ± 0.955	0.063 ± 0.015	2.188 ± 0.891
ConvNeXt-Tiny / CelebA	0.064 ± 0.018	1.019 ± 0.199	0.059 ± 0.017	1.039 ± 0.207
ConvNeXt-Tiny / MetaShift	0.216 ± 0.134	1.144 ± 0.066	0.216 ± 0.172	0.966 ± 0.123

Table 19: Extra negative flips in the damage cohort and initially spared cohort, in pp. Entries are signed $C-D$ differences, computed within paired runs before averaging. Mean $\pm$ SD across original models.

Architecture / dataset	$C_A - D_A$ (pp)	$C_S - D_S$ (pp)	L (pp)
ResNet-18 / Waterbirds	0.393 ± 0.034	0.050 ± 0.067	0.443 ± 0.100
ResNet-18 / CelebA	0.063 ± 0.016	0.060 ± 0.040	0.122 ± 0.054
ResNet-18 / MetaShift	0.699 ± 0.268	0.305 ± 0.264	1.004 ± 0.117
DenseNet-121 / Waterbirds	0.111 ± 0.105	0.253 ± 0.236	0.364 ± 0.284
DenseNet-121 / CelebA	0.027 ± 0.016	-0.017 ± 0.010	0.011 ± 0.009
DenseNet-121 / MetaShift	0.203 ± 0.134	0.064 ± 0.324	0.267 ± 0.413
ConvNeXt-Tiny / Waterbirds	0.015 ± 0.013	0.086 ± 0.068	0.102 ± 0.055
ConvNeXt-Tiny / CelebA	0.005 ± 0.006	-0.021 ± 0.021	-0.016 ± 0.017
ConvNeXt-Tiny / MetaShift	0.000 ± 0.038	0.178 ± 0.117	0.178 ± 0.144

For $K \in \{\mathcal{A}, \mathcal{S}\}$, write C_K, D_K for the corresponding negative-flip contributions. Let W_K count examples in K wrong under both endpoints, U_K those wrong only under C, and V_K those wrong only under D, each as a percentage of the whole test set. Then $C_K = W_K + U_K$, $D_K = W_K + V_K$, and $C_K - D_K = U_K - V_K$. Table 20 gives these joint events for the worked example. The subtraction cancels common errors and retains the difference between the two types of unique errors.

Table 20: Joint endpoint errors for ResNet-18/Waterbirds, original training data and fixed endpoints. All entries are percentages of the whole test set, mean ± SD across three original models, each averaging three recovery repeats. The difference between the final two columns gives the extra negative flips in each cohort.

Cohort	Both wrong	Only C wrong	Only D wrong
Damage cohort $\mathcal{A}$	0.863 ± 0.070	0.545 ± 0.030	0.151 ± 0.063
Initially spared cohort $\mathcal{S}$	1.199 ± 0.177	0.882 ± 0.171	0.832 ± 0.116

C.4 Repair overlap and damage-conditioned repair

Let $b_i \in \{0, 1\}$ indicate correctness under B. The repair-overlap rate in Figure 4 is

$$\frac{\#\{i : o_i = 1, p_i = 0, c_i = 1, b_i = 1\}}{\#\{i : o_i = 1, p_i = 0, c_i = 1\}}. \tag{11}$$

It measures the fraction of predictions repaired by C that are also correct under B. The ratio is computed within each original model and recovery repeat, then averaged using the same aggregation as the other rates. Its denominator is the subset of $\mathcal{A}$ repaired by C; damage-conditioned repair uses all of $\mathcal{A}$. Table 21 gives the cohort sizes and repair-overlap denominators.

Table 21: Cohort sizes for the original-data, fixed-endpoint BatchNorm comparison. N is the whole-test size. Damage count is $\#\{O:1, P:0\}$; the repair-overlap denominator is $\#\{O:1, P:0, C:1\}$. Ranges span three original models with three recovery repeats each. Every repair-overlap denominator is positive.

Architecture	Dataset	N	Damage count	Overlap count
DenseNet-121	CelebA	19962	134–169	97–133
DenseNet-121	MetaShift	874	22–66	14–57
DenseNet-121	Waterbirds	5794	152–215	102–159
ResNet-18	CelebA	19962	168–249	96–164
ResNet-18	MetaShift	874	55–70	33–58
ResNet-18	Waterbirds	5794	158–292	65–220

Table 22: Damage-conditioned repair and repair overlap, original-data fixed endpoints. Both $r_{\mathcal{A}}(B)$ and $r_{\mathcal{A}}(C)$ use the damage cohort $\mathcal{A} = \{O:1, P:0\}$ as denominator; repair overlap uses its subset repaired by C. Values are percentages, mean ± SD across three original models after averaging three recovery repeats. B and C are parallel branches from P.

Architecture	Dataset	$r_{\mathcal{A}}(B)$ (%)	$r_{\mathcal{A}}(C)$ (%)	Overlap (%)
DenseNet-121	CelebA	37.86 ± 6.52	73.61 ± 1.49	26.15 ± 6.54
DenseNet-121	MetaShift	62.31 ± 7.19	79.69 ± 12.86	68.43 ± 10.75
DenseNet-121	Waterbirds	88.03 ± 8.73	70.71 ± 5.86	88.59 ± 7.72
ResNet-18	CelebA	65.83 ± 4.47	63.00 ± 3.88	56.85 ± 5.58
ResNet-18	MetaShift	61.56 ± 17.78	76.20 ± 9.87	73.49 ± 17.08
ResNet-18	Waterbirds	28.00 ± 9.23	56.63 ± 13.66	14.06 ± 5.69

C.5 BatchNorm statistics update

For each applicable P, clone the model and freeze every learned parameter, including BatchNorm affine parameters. Reset running means, running variances, and their counters. Put only BatchNorm modules in training mode, leaving all other modules in evaluation mode. Draw 32 batches of 32 examples uniformly with replacement from the recovery-data regime, with its training augmentation. The statistics-update seed is $s_{\text{base}} + 50000$, using the base seeds in Table 4; it determines the example draws and training augmentations. Run FP32 forward passes and cumulative averaging of the minibatch statistics (BatchNorm momentum set to `None`), then switch back to evaluation mode. A single B is constructed for each original model and recovery-data regime and shared across that model's recovery repeats.

The procedure averages minibatch means and variances over 1,024 sampled positions, which can include repeated examples. The recovered model C starts independently from P and updates its statistics during ordinary recovery training. Thus B and C evaluate two parallel branches from the same pruned model. The BatchNorm statistics update applies to ResNet-18 and DenseNet-121; ConvNeXt-Tiny uses a different normalization architecture.

Table 23: Repair overlap and full-test-set outcomes, original-data regime. The final column counts $p_i = 1, b_i = 0$ as a percentage of the whole test set. SDs are across three original models.

Architecture / dataset	Repair overlap (%)	$\mathrm{Acc}(B) - \mathrm{Acc}(P)$ (pp)	New errors vs. P (%)
ResNet-18 / Waterbirds	14.06 ± 5.69	-16.41 ± 3.18	19.68 ± 3.34
ResNet-18 / CelebA	56.85 ± 5.58	-5.30 ± 0.86	7.14 ± 0.90
ResNet-18 / MetaShift	73.49 ± 17.08	2.17 ± 1.27	3.66 ± 0.30
DenseNet-121 / Waterbirds	88.59 ± 7.72	-17.41 ± 1.18	22.66 ± 1.88
DenseNet-121 / CelebA	26.15 ± 6.54	-3.59 ± 0.69	4.61 ± 0.73
DenseNet-121 / MetaShift	68.43 ± 10.75	0.11 ± 1.32	3.55 ± 0.40

C.6 Trajectory events and margins

At each saved checkpoint t, current correctness is evaluated on a fixed cohort. Figure 5 uses the damage cohort $\mathcal{A}$. The supplementary trajectories also follow $\mathcal{E} = \{i : o_i = 0, p_i = 0\}$, the persistent error cohort: examples wrong under both O and P. Both paths use the same cohort membership within each comparison.

For path M, a renewed error at t means that the prediction was correct in the initial state of that path or at an earlier saved checkpoint, and is wrong at t. Renewed-error rates use the whole test set as denominator. On $\mathcal{A}$, D begins correct and C begins wrong; a renewed error under C therefore requires prior repair. On $\mathcal{E}$, both paths begin wrong and require prior correction. An example meeting the definition at multiple saved checkpoints is counted at each of those checkpoints.

For logits z and z' on the same input, let $\epsilon = \|z' - z\|_\infty$. For true label y, the true-class margin $m = z_y - \max_{k \neq y} z_k$ is positive for a strictly correct prediction, and a negative margin implies an error; m' denotes the margin under z'. It obeys $|m' - m| \leq 2\epsilon$, since the true-class logit and maximum competing logit each change by at most ϵ. Thus $m > 2\epsilon$ is sufficient to retain a strictly correct prediction under this logit change. Exact ties follow the implementation's argmax rule.

D Training randomness and statistical reporting

For one frozen original model and R recovery repeats, let C_r, D_r be the models obtained with seed $r \in \{1, \ldots, R\}$. For classifiers M_1, M_2, label disagreement is $d(M_1, M_2) = N^{-1} \sum_i \mathbf{1}[f_{M_1}(x_i) \neq f_{M_2}(x_i)]$. Between-path disagreement compares C_r with D_s; within-path disagreement compares C_r with C_s or D_r with D_s. Using distinct seeds $r \neq s$, we report

$$\widehat{d}^{\neq}_{CD} = \frac{1}{R(R-1)} \sum_{r \neq s} d(C_r, D_s), \tag{12}$$

$$\widehat{d}_{CC} = \frac{1}{R(R-1)} \sum_{r \neq s} d(C_r, C_s), \qquad \widehat{d}_{DD} = \frac{1}{R(R-1)} \sum_{r \neq s} d(D_r, D_s), \tag{13}$$

$$\widehat{S} = \widehat{d}^{\neq}_{CD} - \tfrac{1}{2}(\widehat{d}_{CC} + \widehat{d}_{DD}). \tag{14}$$

The restriction $r \neq s$ compares distinct seeds. Same-seed pairs C_r, D_r share example draws and augmentation seeds, so their disagreement describes the coupled training paths.

Let $\boldsymbol{\pi}^C_i$ and $\boldsymbol{\pi}^D_i$ be the predicted-label probability vectors at input i across independently sampled training seeds. Their kth entries are the probabilities that the respective path predicts class k. Expected between-path disagreement is $1 - (\boldsymbol{\pi}^C_i)^\top \boldsymbol{\pi}^D_i$, and within-path disagreement is $1 - \|\boldsymbol{\pi}^C_i\|_2^2$ or $1 - \|\boldsymbol{\pi}^D_i\|_2^2$. Consequently,

$$S = \frac{1}{2N} \sum_i \|\boldsymbol{\pi}^C_i - \boldsymbol{\pi}^D_i\|_2^2. \tag{15}$$

Maximum mean discrepancy measures the distance between distributions through a kernel (Gretton et al., 2012). For the categorical equality kernel $k(a, b) = \mathbf{1}[a = b]$ on class labels a, b, S is half its squared value, averaged across inputs. We retain the signed finite-repeat estimate, which can be negative. Positive setting means describe differences between the paths' label distributions relative to sampled within-path variation.

Table 24: Between-path label disagreement minus average within-path label disagreement, $100\widehat{S}$ (pp), using distinct seeds and fixed endpoints. Mean $\pm$ SD across three original models, each with three recovery repeats. Both recovery-data regimes are reported for every setting.

Architecture / dataset	Original training set	Repair subset
ResNet-18 / Waterbirds	0.970 ± 0.147	1.068 ± 0.163
ResNet-18 / CelebA	0.124 ± 0.021	0.347 ± 0.086
ResNet-18 / MetaShift	1.398 ± 0.044	2.962 ± 0.306
DenseNet-121 / Waterbirds	0.481 ± 0.020	0.464 ± 0.043
DenseNet-121 / CelebA	0.045 ± 0.024	0.183 ± 0.067
DenseNet-121 / MetaShift	0.636 ± 0.455	0.826 ± 0.058
ConvNeXt-Tiny / Waterbirds	0.067 ± 0.033	0.012 ± 0.017
ConvNeXt-Tiny / CelebA	0.047 ± 0.009	0.095 ± 0.027
ConvNeXt-Tiny / MetaShift	0.178 ± 0.058	0.292 ± 0.134

For metric z_{tr} on original model t and recovery repeat r, first compute $\bar{z}_t = R^{-1} \sum_r z_{tr}$. Report $\bar{z} = 3^{-1} \sum_{t=1}^{3} \bar{z}_t$ and $s = \sqrt{\sum_t (\bar{z}_t - \bar{z})^2/2}$. The three original models are the independent replication units. Comparisons that share continuation references or use alternative endpoints reuse these original models. SD quantifies training variation on the fixed test set.

E CIFAR-100 RESULTS AND REPRODUCIBILITY

Table 25: CIFAR-100/ResNet-18 at fixed endpoints. Accuracy in percent and differences in pp. Mean $\pm$ SD across three original models, each averaging two recovery repeats.

Pruning condition	P	C	D	$C - P$ (pp)	$C - D$ (pp)
Magnitude 10%	72.87 ± 0.53	74.27 ± 0.26	74.33 ± 0.19	1.40 ± 0.33	-0.06 ± 0.07
Magnitude 20%	60.68 ± 0.35	74.05 ± 0.25	74.33 ± 0.19	13.37 ± 0.42	-0.27 ± 0.06
Random 10%	66.48 ± 1.61	74.06 ± 0.07	74.33 ± 0.19	7.58 ± 1.67	-0.26 ± 0.18
Random 20%	39.74 ± 7.81	73.81 ± 0.17	74.33 ± 0.19	34.07 ± 7.65	-0.52 ± 0.35

Table 26: Extra corrections and extra negative flips relative to matched continuation. All four CIFAR-100 conditions use cross-entropy in original training, recovery, and continuation. Differences are in pp, mean $\pm$ SD across three original models, each averaging two recovery repeats. $G + L$ is the difference in correctness turnover relative to O.

Pruning condition	G (pp)	L (pp)	$G - L$ (pp)	$G + L$ (pp)
Magnitude 10%	0.223 ± 0.108	0.283 ± 0.045	-0.060 ± 0.074	0.507 ± 0.148
Magnitude 20%	0.305 ± 0.080	0.577 ± 0.018	-0.272 ± 0.063	0.882 ± 0.098
Random 10%	0.450 ± 0.187	0.712 ± 0.126	-0.262 ± 0.176	1.162 ± 0.266
Random 20%	0.863 ± 0.171	1.382 ± 0.208	-0.518 ± 0.346	2.245 ± 0.160

E.1 TRAJECTORIES IN THE DAMAGE COHORT AND PERSISTENT ERROR COHORT

The saved predictions underlying Table 25 provide trajectories for all four pruning conditions. Figure 5 tracks current correctness in $\mathcal{A}$. Figure 6 tracks current correctness in $\mathcal{E}$, and Figures 7 and 8 give renewed-error rates for the two cohorts.

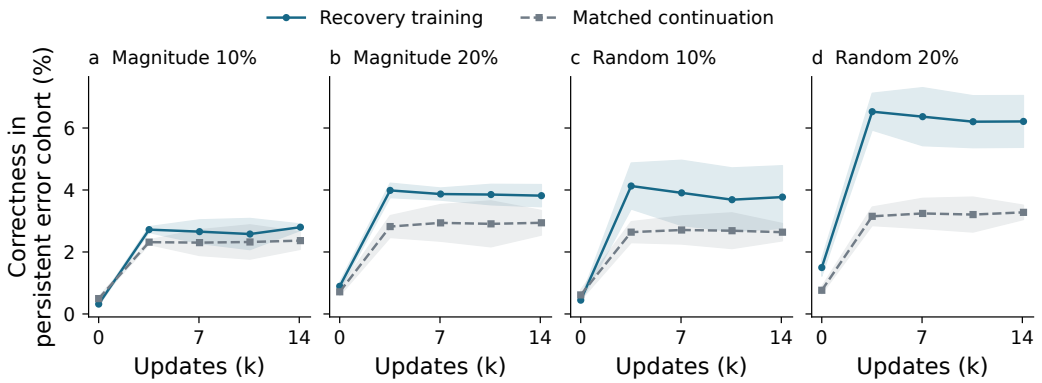


Figure 6: Current correctness in the persistent error cohort $\mathcal{E}$. Both paths start incorrect on this cohort. Bands are SD across three original models after averaging two recovery repeats, for the same four conditions as Figure 5.

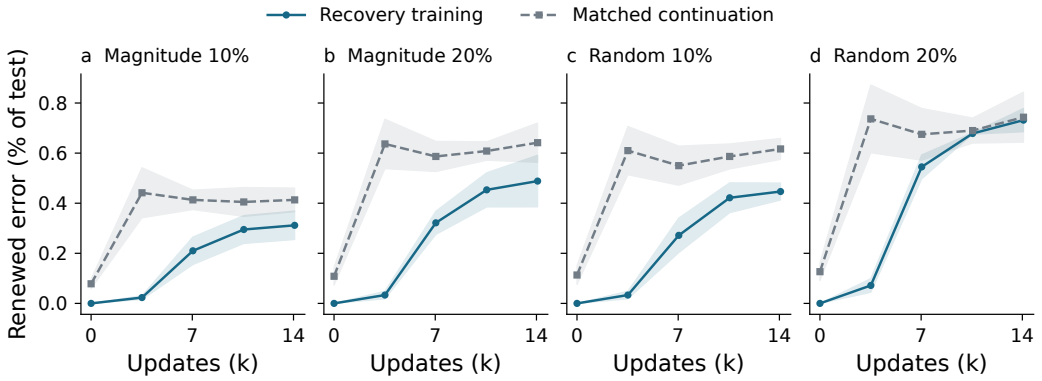


Figure 7: Renewed errors in the damage cohort $\mathcal{A}$, as a percentage of the whole test set. For D, prior correctness includes O at update zero. For C, it requires repair at an earlier saved checkpoint. The rate therefore depends on both prior correctness and its subsequent loss. Bands use the same aggregation across original models.

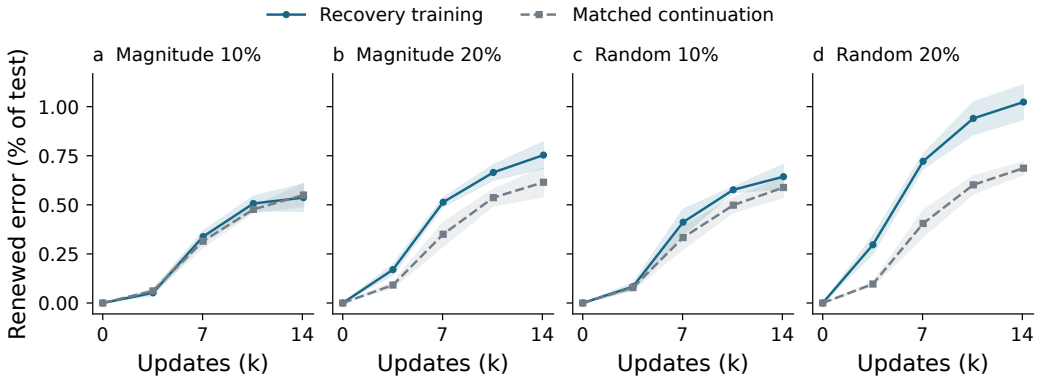


Figure 8: Renewed errors in the persistent error cohort $\mathcal{E}$, as a percentage of the whole test set. Both paths require prior correction to enter this event. Renewed errors occur in both paths, with differences across pruning conditions.

The analysis source contains event counts, summaries for each original model, hashes identifying protocol configurations, figure scripts, and table-generation code. Image datasets and model assets are obtained from their upstream sources under the corresponding licenses.

F SUPPLEMENTARY DETAILS FOR THE RETENTION DIAGNOSTIC

Section 5.7 defines the continuation-referenced retention diagnostic R_{keep} and its proposed evaluation uses. Here we expand its relation to net losses, formalize the constrained comparison, and specify how to test its selection value.

Relation to net losses. The existing net difference can be written as

$$L = \widehat{\Pr}(o_i = 1, d_i = 1, c_i = 0) - \widehat{\Pr}(o_i = 1, d_i = 0, c_i = 1). \tag{16}$$

The two terms can offset. Dividing the first term by $\widehat{\Pr}(o_i = 1, d_i = 1)$ gives $1 - R_{\mathrm{keep}}$, the fraction of predictions correct under both O and D that C loses. Restricting the sums in Eq. 7 to a predefined group or to $\mathcal{A}$ or $\mathcal{S}$ gives the corresponding conditional rate. Report each denominator and treat empty reference sets as undefined. This diagnostic measures retention at an endpoint; Section 5.6 provides the complementary trajectory analysis.

Accuracy and retention requirements. Accuracy measures aggregate predictive utility when errors have equal cost. Tasks can additionally specify limits on losses of previously correct predictions. These limits define a retention requirement alongside overall and group accuracy. Absolute group accuracy and negative flips relative to O expose losses shared by C and D. At fixed computation, Section 5.7 proposes ranking checkpoints that meet the accuracy requirements by R_{keep}.

For the comparison across pruning levels, let $\mathcal{K}$ be a finite set of candidates from the same O, evaluated against a fixed continuation endpoint D. Write $s(C) = 1 - \mathrm{FLOPs}(C)/\mathrm{FLOPs}(O)$. The largest tested saving satisfying the specified requirements is

$$S^*(\varepsilon) = \max_{C \in \mathcal{K}} \left\{ s(C) : \mathrm{Acc}(C) \geq a_{\min},\ \mathrm{Acc}_g(C) \geq a_{g,\min}\ \forall g,\ R_{\mathrm{keep}}(C; O, D) \geq 1 - \varepsilon \right\}. \tag{17}$$

The accuracy floors and tolerated regression $\varepsilon \in [0, 1]$ encode task-specific requirements. Apply group constraints as specified by the task. If none qualify, report no feasible candidate. The criterion reports the largest qualifying FLOP saving among the tested candidates. Energy use requires separate hardware measurements.

Validation protocol. Fix O, the continuation endpoint D, the candidate set, and thresholds before validation-based selection. Use the same D across candidates to keep the reference set constant. Candidate checkpoints can differ in their realized update counts; this selection comparison evaluates them against a shared continuation endpoint. Compare against accuracy, worst-group accuracy, and NFR-based selection under the same requirements and candidate set.

Evaluate selected models on untouched data and independent training repeats. Report threshold sensitivity, cohort sizes, and conditional rates computed within each paired repeat before aggregation. Same-path continuation repeats provide a descriptive estimate of training variation. Accuracy and retention rates are sample estimates; report uncertainty when assessing whether the requirements are met. Assess selection value through the selected models' held-out accuracy, group performance, and prediction retention relative to the baselines. This protocol tests the selection benefits suggested by the present findings.