The capacity audit: most class-incremental routing error is placement slack, not capacity

W. Schulz

Jul 30, 2026

Abstract

Is the router-error plateau in class-incremental learning a sphere-packing capacity limit? Two floor estimators were killed by their own pre-set validity checks. A validated simulator shows packing stops binding near 16 dimensions; at deployment dimensions roughly 80% of the error is placement slack.

A short chapter about a small number, and about three instruments that had to die before it could be measured.

The continual-learning track had established that class-incremental forgetting is not a loss of knowledge but a loss of *selection* — the individual experts stay near-perfect while the router that picks between them carries the entire loss. That result made the router's error rate the lever, and a follow-up campaign found that the error-versus-diversity curve flattens into a plateau which looks exactly like a capacity bound: a d-dimensional space holds only so many well-separated classes, so past some point you are fighting sphere-packing, and the only remedy is more dimensions.

If that is right, it decides where the engineering effort should go. So the question was made quantitative: how much of the achieved error is capacity, and how much is placement slack? Take the real embeddings — a vision transformer on a hundred-class benchmark, already on disk — measure the achieved routing error, compute the error a *near-optimal packing* of the same number of classes at the same measured noise level would achieve, and subtract.

Three instruments, two killed by their own gates

The first attempt built the floor analytically, from a pairwise error law plus a union bound over rivals plus a worst-case packing inequality. It died on its own registered gate: it overpredicts measured error by a factor of 2.1 to 2.35 at high dimension, and at low dimension the computed "floor" sits *above* the achieved error — a floor above the ceiling. The construction that puts every rival simultaneously at minimum distance is a worst case, not an optimum. No headroom numbers were drawn from that run.

The second attempt built the floor by transplant — take the measured residuals, place them on a constructed packing, and measure directly. It died too, on a pre-committed kill, for a subtler reason: residuals returned to their own centroids reconstruct the training points, so the measurement was in-sample, reading 0.017 where the true test error was 0.307. That gap is itself a striking observation about nearest-centroid classifiers, and it was recorded as one.

But that run carried a registered diagnostic column, and the column validated: an isotropic Gaussian simulator, using the *measured* per-class noise, tracked real test error to within a factor of 0.94–1.16 at

every dimension. So the third instrument was that simulator, promoted with a fresh-seed kill gate — and it validated again, 0.90 to 1.18 across the full dimension range, then passed 5 of 5 on the audit itself.

The answer

dimension	achieved error	achievable floor	headroom
8	0.821	0.515	0.33
16	0.670	0.199	0.67
32	0.504	0.139	0.72
128	0.352	0.075	0.80
768	0.306	0.067	0.81

Packing stops binding at about sixteen dimensions. At deployment dimensions roughly 80% of the achieved error is placement slack above a floor that is achievable by construction. Going from 128 dimensions to 768 buys 0.046 of error while at least 0.80 of slack persists at both ends.

So the plateau is not a capacity wall. Routing and placement quality, not capacity, is the lever at deployment scale — now with a floor-certified number underneath the earlier finding that better routing and doing work at write time can bend the curve.

What this does *not* say, and the file says so explicitly: that any trainable method reaches the floor. A discriminative placement method already demonstrated in this track that it forgets under streaming. The floor is a capacity statement, not a training recipe.

And a retraction, executed by a pre-committed rule

The first, killed instrument had produced one flag that looked substantive: the shipped learned-key lever collapses within-class spread ninefold while *worsening* the nearest-centroid error, which would be the signature of contrastive collapse and a reason to stop quoting learned keys as a geometry improvement.

That flag was registered with a kill-for-flag clause, and the clause fired. At the original protocol the lever improves nearest-centroid error from 0.369 to 0.330 while the spread still collapses more than sixfold, and at equal dimension it beats a random-projection baseline 0.371 against 0.420. The worsening was a training artifact of a different sample size, and it is withdrawn; the single anomalous observation stands as unexplained and nothing more.

The method lesson recurred three times in one day across three unrelated strands, which is why it is written here rather than in a footnote: analytic floors built from worst-case bounds and independence approximations are not floors on real geometry. Registered instrument-validity gates are what keep a run like this from generating confident wrong numbers — and in this case they killed two instruments before letting the third speak.

Sources

Every number in this chapter (the three instruments, the headroom table, the learned-key retraction) is this project's own run-once measurement, recorded in `continual-learning/` *in the Quantum-Intelligence repository.*

1. Kim, G., Xiao, C., Konishi, T., Ke, Z., & Liu, B. (2022). "A Theoretical Study on Solving Continual Learning." *NeurIPS 2022.* arXiv:2211.02633 — the published decomposition of class-incremental accuracy into within-task prediction and task-identity prediction, which the selection framing this chapter starts from restates (see the ε correction).

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/capacity-audit