
The Spectral Anatomy of Plasticity Loss

Mahdi Kallel

University of Würzburg, Germany
mahdi.kallel@uni-wuerzburg.de

Nour Jamoussi

EURECOM, France
nour.jamoussi@eurecom.fr

Johannes Tölle

University of Luxembourg, Luxembourg
johannes.toelle@uni.lu

Marios Kountouris*

DaSCI, University of Granada, Spain
mariosk@ugr.es

Carlo D’Eramo*

University of Würzburg, Germany
carlo.deramo@uni-wuerzburg.de

Abstract

Continual learning can make a network progressively worse at fitting new tasks within a fixed training budget. The mechanism behind this loss of plasticity remains unclear. We leverage random matrix theory to follow learned structure as it emerges from the random bulk of the network’s weight matrices. Across an MLP, ResNet-18, and ViT-Tiny, we show that persistent upper-edge spectral structure accumulates over training and remains functionally important at later checkpoints. We further find in ResNet-18 and ViT-Tiny that later net parameter displacement becomes increasingly concentrated in pre-existing spectral-edge directions. At a representative ViT-Tiny checkpoint, the same upper-edge subspace is enriched in both input covariance and backpropagated error, exposing it to the task on both sides of the layer.

1 Introduction

Previously trained neural networks do not remain equally trainable. Given the same new task and the same number of updates, a previously trained network can reach a lower training accuracy than a fresh network. This growing gap is loss of plasticity. It limits how long a model can continue fitting new tasks under a fixed training budget [7, 2, 13].

Prior work has connected plasticity loss to changes in activations [1], feature subspaces [16], optimizer dynamics [3], weight spectra [5, 12, 14], and loss curvature [7, 6]. A particularly relevant result is the following. As networks lose plasticity, curvature and per-example gradients on a new task span fewer independent directions [6, 5, 12]. This identifies a contraction in gradient and curvature dimensionality, but not how that contraction relates to structure in the trained weights. We ask whether learned spectral structure directs later gradients into a progressively smaller set of directions.

Random matrix theory (RMT) [11] gives us a way to locate this structure. The singular values of a random weight matrix form a predictable bulk. Training produces departures from this bulk, and the corresponding singular vectors provide candidate learned directions. Earlier work has used these departures to locate information in trained networks and has shown that both spectral edges can matter [10, 15, 14]. We extend this picture over time. We follow these directions across tasks, test what function they carry, and measure how later learning moves through them.

*Equal senior contribution.

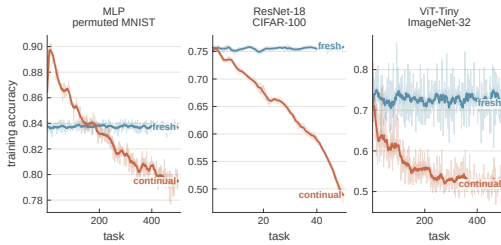


Figure 1: **Plasticity declines with model age.** Training accuracy on the current task for continual and fresh models across the three settings; the fresh-minus-continual gap measures loss of plasticity under a fixed update budget.

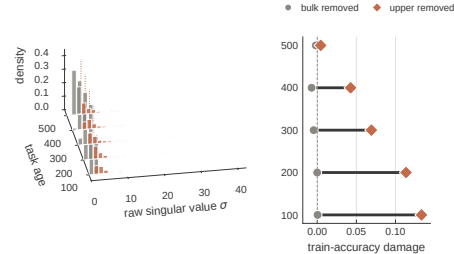


Figure 2: **Training produces upper edges beyond the random bulk.** Left: ViT-Tiny spectra across tasks. Right: training-accuracy decrease after removing matched numbers of upper-edge or bulk singular components.

We study loss of plasticity in an MLP on permuted MNIST, a ResNet-18 on label-permuted CIFAR-100, and a ViT-Tiny on a binary ImageNet-32 task stream by tracking weight spectra, retained function, checkpoint displacement, and task-signal statistics across successive tasks. We find that continual learning builds persistent upper-edge spectral structure that accumulates over training and remains functionally important at later checkpoints. As plasticity declines in ResNet-18 and ViT-Tiny, net parameter displacement becomes increasingly concentrated in spectral-edge directions already present at the beginning of each interval. Finally, we find that in ViT-Tiny, the upper-edge subspace is enriched in both input-side covariance and backpropagated error, showing that this spectral region is exposed to the task on both sides of the layer.

2 Weight Spectra and Random Matrix Theory

We analyze each weight matrix $W \in \mathbb{R}^{m \times n}$ through its compact singular value decomposition

$$W = U\Sigma V^\top = \sum_{i=1}^p \sigma_i u_i v_i^\top, \quad p = \min(m, n).$$

RMT provides a reference for the weight spectrum before learning. Let $N = \max(m, n)$ and $\gamma = p/N \leq 1$. For a random matrix with independent, centered entries of variance $1/N$, the empirical distribution of squared singular values σ^2 converges, in the large-dimensional limit, to the Marchenko–Pastur (MP) law [9], whose support is $[\lambda_-, \lambda_+]$, with $\lambda_{\pm}(1 \pm \sqrt{\gamma})^2$. Neural weight matrices at random initialization are often well described by this random-matrix model. At finite width, however, the extreme squared singular values fluctuate around the asymptotic limits λ_- and λ_+ and can therefore lie slightly outside the MP bulk. At soft spectral edges, Tracy–Widom theory characterizes the scale of these finite-size fluctuations; the lower edge requires additional care when the aspect ratio approaches one and the MP lower edge approaches zero [4, 8]. Training changes the spectrum. When a singular direction lies beyond the range predicted by the fitted random-matrix bulk, we treat it as candidate learned spectral structure, while allowing for both outliers that emerge during training and finite-size outliers already present at initialization. We call these directions the spectral edges. Directions above the bulk form the upper edge, while directions below it form the lower edge [10, 15]. Staats et al. [14] show that directions at both ends of transformer weight spectra can align with the data. Small singular directions below the bulk can also be important for the model function. We therefore measure the lower and upper edges in the same way.

In the following section, we track how this learned spectral structure accumulates across tasks, what earlier knowledge it retains, and how subsequent parameter displacement is expressed through it.

3 Empirical Analysis

We consider three continual-learning settings: an MLP on permuted MNIST, a ResNet-18 on label-permuted CIFAR-100, and a ViT-Tiny on binary ImageNet-32 tasks. We call the number of tasks

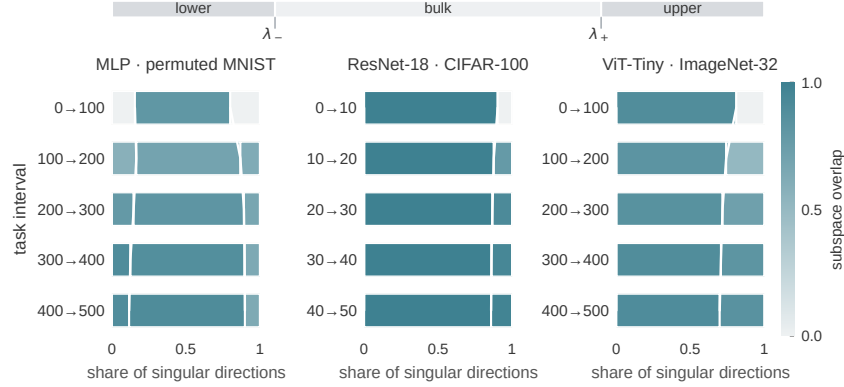


Figure 3: **Upper-edge directions become increasingly persistent across training.** Each row compares consecutive task intervals. Width shows the share of singular directions assigned to the lower edge, MP bulk, and upper edge; color indicates consecutive subspace overlap. The overlap statistic and its normalization are defined in Appendix A.

already seen the model’s age. At selected ages, we train the continual model and a fresh network on the same new task, using the same examples, minibatch order, optimizer type and hyperparameters, and number of updates. The continual model retains optimizer state, whereas the fresh model starts with a new optimizer; see Appendix A.1. We define the plasticity gap as $A_{\text{fresh}} - A_{\text{continual}}$, so a positive value means that the previously trained model is harder to fit within the same budget. Each selected checkpoint is decomposed into an MP bulk and lower- and upper-edge according to the RMT criterion above. Appendix A provides the full training and measurement details.

Learned structure accumulates above the random bulk. Figure 2 shows the ViT-Tiny weight spectrum across model ages. As training proceeds, singular values increasingly extend beyond the upper edge of the bulk, forming a distinct population of upper-edge outliers. These directions are also functionally relevant. Removing upper-edge components causes substantially more training-accuracy damage than removing the same number of bulk components at every displayed age. The effect is strongest at earlier ages, with an accuracy drop of roughly 0.13 at age 100, and gradually decreases as the model ages. Thus, the upper spectral edge does not merely depart from the random-matrix prediction: upper-edge components are more functionally consequential per removed direction than rank-matched bulk components. Because rank matching generally removes more Frobenius energy from upper-edge components, this experiment localizes function but is not by itself an energy-matched causal comparison; Appendix A describes how energy-matched bulk controls are selected when feasible. We therefore view the upper edge as a persistent spectral record accumulated over training. Results for the MLP and ResNet-18 are reported in Appendix B. We next ask how this structure evolves as plasticity declines.

Upper-edge directions stabilize as plasticity falls. Figure 3 tracks the lower edge, MP bulk, and upper edge across consecutive task intervals, together with the overlap of each subspace between checkpoints. Across the three settings, the upper edge becomes progressively more persistent: its share of the spectrum increases, while consecutive upper-edge subspaces show high overlap at later ages. This stabilization occurs over the same period in which the plasticity gap grows. These observations suggest that upper-edge structure becomes increasingly stable rather than being fully replaced between checkpoints as the network becomes harder to train.

Later parameter displacement concentrates in existing spectral-edge directions. We examine how net parameter displacement over each task interval is expressed in the spectral structure already present at the interval’s starting checkpoint. Using the frozen source singular basis, we separate the in-frame displacement into EE , EB , BE , and BB blocks according to whether its left and right singular directions belong to the complete spectral edge $E = E^- \cup E^+$ or to the MP bulk. We refer to EE , EB , and BE together as the edge-touching route. The net checkpoint displacement progressively shifts toward the existing spectral edge (Figure 4). In ResNet-18, the share of movement

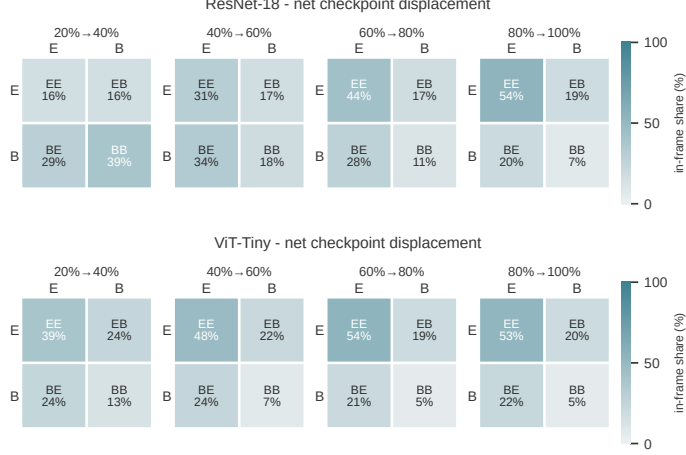


Figure 4: **Net checkpoint displacement progressively shifts toward existing spectral-edge directions.** From top to bottom: ResNet-18 on CIFAR-100, and ViT-Tiny on ImageNet-32. Each panel reports the net checkpoint displacement in the singular basis saved at the beginning of each interval.

in the EE block increases from 16% in the earliest interval to 54% in the latest, while the BB share drops from 39% to 7%. ViT-Tiny shows the same redistribution: EE rises from 39% to approximately 53%, whereas BB decreases from 13% to 5%. The mixed EB and BE blocks remain substantial, indicating that later parameter displacement is not confined entirely to the edge, but is increasingly expressed through directions that already belong to the learned spectral structure. Because E combines lower- and upper-edge directions, this result establishes increasing concentration in the complete spectral edge; it does not by itself attribute the increase specifically to the upper edge. Results with MLP on permuted MNIST are reported in Appendix B. This shift motivates a closer look at the upper-edge directions themselves and at whether they are engaged by the task on both sides of the layer.

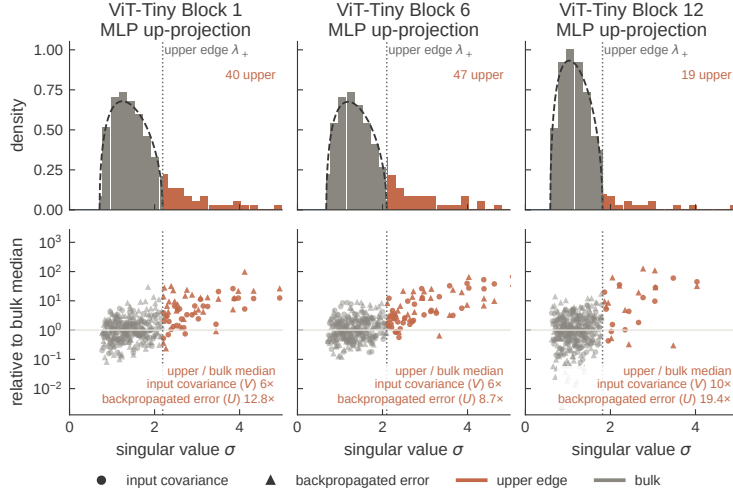


Figure 5: **Upper-edge directions are exposed on both sides of the MLP projection.** Top: empirical singular-value spectra of the 768×192 ViT-Tiny MLP up-projections in blocks 1, 6, and 12 at task 100, together with the MP bulk. Bottom: input-covariance energy $v_k^T C_h v_k$ and backpropagated-error second moment $u_k^T M_\delta u_k$ for each singular direction, normalized by the bulk median of the same layer. The upper-edge subspace is enriched in both quantities; no lower-edge outliers are detected.

The upper-edge subspace carries both forward and backward task signal. Following the RMT principle that departures from the random bulk may identify learned directions, we ask whether

upper-edge singular vectors are exposed to the task on both sides of the layer. Figure 5 shows that at the task-100 checkpoint, upper-edge directions of the ViT feed-forward up-projection carry $5.65\times$, $5.58\times$, and $9.73\times$ the bulk-median input-covariance energy in blocks 1, 6, and 12, respectively. Their corresponding backpropagated-error second moments are enriched by $12.75\times$, $8.66\times$, and $19.41\times$. Thus, the same non-random spectral region is prominent both in the representations entering the matrix and in the error signal returning through it. This correspondence between input covariance and backpropagated error is observational: it localizes late-task traffic to the upper edge, but does not by itself establish that the edge causes plasticity loss.

4 Conclusion

We studied loss of plasticity through the spectral structure of neural network weight matrices. Across an MLP on permuted MNIST, a ResNet-18 on label-permuted CIFAR-100, and a ViT-Tiny on binary ImageNet-32 tasks, we found that training builds persistent upper-edge structure that accumulates over training and remains functionally important at later checkpoints. Across all three settings, upper-edge directions become increasingly persistent with model age, while in ResNet-18 and ViT-Tiny net checkpoint displacement becomes increasingly concentrated in the complete spectral edge. These results are consistent with later optimization becoming increasingly organized around spectral directions that are already present, rather than establishing that gradient magnitude vanishes or that the upper edge itself causes plasticity loss. In ViT-Tiny, the upper-edge subspace is enriched in both input covariance and backpropagated error, indicating that late-task signal is organized around spectral structure established earlier in training.

Acknowledgments

Marios Kountouris was partially supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement No. 101003431) and by the SNS JU under Horizon Europe (Grant No. 101192080, 6G-LEADER). Nour Jamoussi was supported by the IMT “Futur, Ruptures & Impacts” programme.

References

- [1] Daniel Anthes, Sushrut Thorat, Peter König, and Tim C Kietzmann. Keep moving: identifying task-relevant subspaces to maximise plasticity for newly learned tasks. In Vincenzo Lomonaco, Stefano Melacci, Tinne Tuytelaars, Sarath Chandar, and Razvan Pascanu, editors, *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pages 815–831. PMLR, 29 Jul–01 Aug 2025.
- [2] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.
- [3] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.
- [4] Shira Kritchman and Boaz Nadler. Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory. *IEEE Transactions on Signal Processing*, 57(10):3930–3941, 2009.
- [5] Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, Andras Gyorgy, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C Machado. Learning continually by spectral regularization. In *International Conference on Learning Representations*, volume 2025, pages 34504–34530, 2025.
- [6] Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2023.
- [7] Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pages 23190–23211. PMLR, 2023.

- [8] Zongming Ma. Accuracy of the Tracy—Widom limits for the extreme eigenvalues in white wishart matrices. *Bernoulli*, 18(1):322–359, 2012.
- [9] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.
- [10] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.
- [11] Madan Lal Mehta. *Random Matrices*, volume 142. Elsevier, 2004.
- [12] Arjun Prakash, Naicheng He, Kaicheng Guo, Saket Tiwari, Ruo Yu Tao, Tyrone Serapio, Amy Greenwald, and George Konidaris. Spectral collapse drives loss of plasticity in deep continual learning. *arXiv preprint arXiv:2509.22335*, 2025.
- [13] Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pages 32145–32168. PMLR, 2023.
- [14] Max Staats, Matthias Thamm, and Bernd Rosenow. Small singular values matter: A random matrix analysis of transformer models. *Advances in Neural Information Processing Systems*, 38:153545–153573, 2025.
- [15] Matthias Thamm, Max Staats, and Bernd Rosenow. Random matrix analysis of deep neural network weight matrices. *Physical Review E*, 106(5):054124, 2022.
- [16] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3718–3727. IEEE, 2023.

A Experimental Setup

A.1 Overview

We evaluate the relation between spectral consolidation and loss of plasticity in three continual-learning settings comprising a multilayer perceptron on permuted MNIST, a ResNet-18 on label-permuted CIFAR-100, and a ViT-Tiny on binary ImageNet-32 tasks. Each of these experiments contains a sequence of tasks trained with a fixed per-task optimization budget. At each evaluated task age we compare a continually trained model with a fresh model trained on the same task. Both models are trained on the identical task data, minibatch order, and update budget. For a continually trained model the optimizer state is continuous across task boundaries. In contrast, fresh reference models are rebuilt from the shared seed-specific initialization and receive a new optimizer for every evaluated task. Because the continual model retains optimizer state whereas the fresh reference resets it, the measured gap reflects the joint effect of the learned parameter state and accumulated optimizer state. For Adam-based ResNet-18 and ViT-Tiny experiments, a continual-weights/optimizer-reset control would be needed to isolate these contributions. For endpoint accuracy, we use the fresh-minus-continual difference so that a positive value denotes loss of plasticity.

For all experiments, we use a basic setup. All main multi-seed experiments use five seeds and the datasets are used without data augmentation. No gradient clipping is applied.

Throughout the paper, we restrict the spectral analysis to hidden Linear layers and exclude convolutional layers, for which the RMT assumptions are less reliable. For the particular case of the CIFAR-style ResNet-18 variant used here we include an additional hidden Linear layer of width 512 after global average pooling, followed by ReLU and a 100-way output classifier. The reported ResNet spectral and checkpoint-routing analyses concern this single 512×512 hidden weight matrix. Convolutional weights and the final classifier remain trainable but are excluded from these measurements.

A.2 Task streams and architectures

Permuted MNIST. The MNIST stream contains 500 tasks. The first task uses the original pixels, and each subsequent task applies a seed-dependent random permutation to the input coordinates while preserving the ten digit labels. Each task contains 10,000 training images, which are mapped to $[-1, 1]$ and flattened before being passed to a two-hidden-layer ReLU MLP. Both hidden layers have width 100, and the output layer has ten units in order to match the size of the label space.

Each task is trained for five epochs using stochastic gradient descent with learning rate 0.01, zero momentum, batch size 16, no weight decay, and no gradient clipping. Checkpoints are retained after 50, 100, 200, 300, 400, and 500 completed tasks.

CIFAR-100 label-vector permutations. The CIFAR-100 stream uses the same 50,000 training images at every task. A task is formed by independently permuting the fixed balanced vector of CIFAR-100 training labels across examples. Consequently, every task has the same inputs and class frequencies but a different input-label association. We generate 50 tasks per seed.

The model is a CIFAR-style ResNet-18 variant with a 3×3 convolutional stem, four residual stages with widths 64, 128, 256, and 512, Batch Normalization, and a global-average-pooling classification head. After global average pooling, the 512-dimensional representation passes through a hidden Linear(512, 512) layer with ReLU activation, followed by a Linear(512, 100) output classifier.

Convolutional and hidden Linear weights use explicit Kaiming initialization. Images are normalized using the CIFAR-100 training-set channel statistics, without augmentation.

We train each task for ten epochs with Adam, learning rate 10^{-4} , batch size 250, zero weight decay, no scheduler, and no warm-up. Checkpoints are retained after tasks 1, 5, 10, 20, 30, 40, and 50.

ImageNet-32 class pairs. For each seed, the 1,000 ImageNet classes are randomly permuted once and divided into 500 disjoint binary tasks. Each task consists of one class pair and contains 600 training examples per class. The 32×32 image tensors are converted to float32 without additional normalization or augmentation. The same continuous two-way classifier is used for every task.

The model is a 12-block ViT-Tiny with embedding width 192, three attention heads of width 64, feed-forward width 768, pre-LayerNorm residual blocks, and a class token. Images are divided into non-overlapping 4×4 patches, producing 64 image tokens plus the class token. Attention uses one fused QKV projection per block, and the attention and embedding dropout probabilities are both 0.1. The final classifier maps the 192-dimensional class-token representation to two logits.

Each binary task is trained for ten epochs with Adam at learning rate 10^{-3} and batch size 100. Adam uses its default momentum coefficients and zero weight decay. No learning-rate scheduler, warm-up, or gradient clipping is used. Checkpoints are retained after tasks 50, 100, 200, 300, 400, and 500.

A.3 Spectral atlas

For each eligible hidden Linear weight matrix $W \in \mathbb{R}^{m \times n}$, we compute the compact singular value decomposition

$$W = U \Sigma V^\top.$$

Let $p = \min(m, n)$, $N = \max(m, n)$, and $\gamma = p/N$. We analyze the squared singular values σ_k^2 .

Bulk-scale estimation. Let s denote the fitted scale of the squared-singular-value bulk. Under the random-matrix reference model, the MP boundaries are

$$\lambda_{\pm} = s(1 \pm \sqrt{\gamma})^2.$$

We initialize s using the median squared singular value. At each iteration, we compute the finite-size thresholds defined below and update s to the mean squared singular value among directions currently assigned to the bulk. Iteration stops when the relative scale change is below 10^{-4} , after 20 iterations, or if fewer than five singular values remain in the bulk. In the latter case, the previous scale estimate is retained.

Finite-size thresholds. The MP boundaries describe an asymptotic random-matrix spectrum. At finite matrix sizes, extreme squared singular values fluctuate around these boundaries even under the random reference model. We therefore use Tracy–Widom-inspired finite-size thresholds rather than the uncorrected MP boundaries.

Writing $r_N = \sqrt{N - \frac{1}{2}}$ and $r_p = \sqrt{p - \frac{1}{2}}$, the upper-edge center and fluctuation scale are

$$\mu_+ = \frac{s}{N}(r_N + r_p)^2, \quad \tau_+ = \frac{s}{N}(r_N + r_p) \left(\frac{1}{r_N} + \frac{1}{r_p} \right)^{1/3}.$$

We set

$$\lambda_+^{\text{TW}} = \mu_+ + c_{\text{TW}} \tau_+, \quad c_{\text{TW}} = 2.$$

The multiplier c_{TW} specifies a margin in units of the finite-size fluctuation scale.

For $\gamma < 1 - 10^{-6}$, the lower threshold is

$$\begin{aligned} \mu_- &= \frac{s}{N}(r_N - r_p)^2, \\ \tau_- &= \frac{s}{N}(r_N - r_p) \left(\frac{1}{r_p} - \frac{1}{r_N} \right)^{1/3}, \\ \lambda_-^{\text{TW}} &= \max\{\mu_- - c_{\text{TW}} \tau_-, 0\}. \end{aligned}$$

For $\gamma \geq 1 - 10^{-6}$, we instead set $\lambda_-^{\text{TW}} = 0$. In particular, square matrices have a zero lower MP boundary: this is a hard edge because squared singular values cannot be negative. We do not apply a lower soft-edge correction in this case.

Compartment assignment and matrix shapes. Singular directions are assigned to

$$\begin{aligned} E^- &= \{k : \sigma_k^2 < \lambda_-^{\text{TW}}\}, \\ B &= \{k : \lambda_-^{\text{TW}} \leq \sigma_k^2 \leq \lambda_+^{\text{TW}}\}, \\ E^+ &= \{k : \sigma_k^2 > \lambda_+^{\text{TW}}\}. \end{aligned}$$

The complete spectral edge is $E = E^- \cup E^+$. Lower and upper edges are measured separately before being combined.

The analyzed matrices include both square and rectangular weights. The MLP uses 100×784 and 100×100 matrices; the ResNet-18 analysis uses its single 512×512 hidden head matrix. Each ViT-Tiny block contributes a 576×192 fused QKV projection, a 192×192 attention output projection, and 768×192 and 192×768 MLP projections. Thus, the aspect ratios are approximately 0.128, $1/4$, $1/3$, or 1: none of the analyzed rectangular matrices is nearly square.

For square matrices, the zero lower threshold implies $E^- = \emptyset$ under this detection rule. Consequently, the combined edge in the ResNet-18 analysis consists only of upper-edge directions.

A.4 Consecutive subspace overlap

For consecutive checkpoints $a < b$ and compartment $C \in \{E^-, B, E^+\}$, let $U_{a,C}$ and $V_{a,C}$ contain the corresponding left and right singular vectors of $W_a = U_a \Sigma_a V_a^\top$, and define $U_{b,C}, V_{b,C}$ analogously. Writing $k_{b,C}$ for the number of directions in compartment C at checkpoint b , we measure

$$O_C(a \rightarrow b) = \frac{\|U_{b,C}^\top U_{a,C}\|_F^2 + \|V_{b,C}^\top V_{a,C}\|_F^2}{2k_{b,C}}.$$

Each squared Frobenius norm sums the squared cosines of the principal angles between the corresponding subspaces. Normalization by the later dimension makes this the average fraction of the current subspace captured by the previous one, equally weighting the input and output sides. The score lies in $[0, 1]$: zero denotes orthogonality, and one denotes containment of the current subspaces in the previous ones. It is asymmetric when dimensions differ and is not corrected for chance overlap. Within each seed, matrix-level scores are weighted by $k_{b,C}$; Figure 3 then averages across seeds. If a compartment is absent at the previous checkpoint but present at the current one, its overlap is zero. Matrices with no current directions in the compartment contribute neither to the numerator nor to the denominator of the within-seed aggregate. If no analyzed matrix contains current directions in that compartment, the seed-level overlap is undefined and excluded from the across-seed mean.

A.5 Functional removal controls

To test whether checkpoint-local upper-edge directions carry learned function, we set selected singular values to zero and reconstruct the matrix using the original singular vectors $\tilde{W} = U\tilde{\Sigma}V^\top$. Upper-edge removals are compared with same-layer central-bulk removals having exactly the same rank. Where an energy-matched control is feasible, the bulk window is chosen to minimize the difference in removed Frobenius energy. Removed rank, removed energy, affected layers, and any saturation of the available bulk pool are recorded. Rank matching alone is treated as functional localization rather than dose-matched causal evidence.

A.6 Frozen spectral routing and checkpoint motion

A source checkpoint a fixes U_a, V_a , and the lower/bulk/upper labels. For any later matrix-valued signal X , we compute

$$J_X = U_a^\top X V_a. \quad (1)$$

The resulting coefficients form a 3×3 grid indexed by the U /error-side and V /activation-side compartments. Because these nine blocks are disjoint,

$$\|X\|_F^2 = \sum_{C,D \in \{E^-, B, E^+\}} \|J_{CD}\|_F^2 + \|X_\perp\|_F^2. \quad (2)$$

Every routing receipt includes the compact-frame coverage, the off-frame energy, and the numerical closure error.

For checkpoint motion from age a to age b , we set

$$X = \Delta W_{a \rightarrow b} = W_b - W_a. \quad (3)$$

This is signed net displacement over an entire task interval. Opposing optimizer steps may cancel, so checkpoint displacement is not interpreted as the sum of per-step update energy.

Aggregation for Figure 4. We combine the lower and upper edges as $E = E^- \cup E^+$. For each seed s and interval $a \rightarrow b$, let

$$J_{s,\ell} = U_{a,s,\ell}^\top (W_{b,s,\ell} - W_{a,s,\ell}) V_{a,s,\ell},$$

where ℓ indexes the analyzed hidden Linear weight matrices, excluding convolutions and the final classifier. For ResNet-18, this sum contains only the 512×512 hidden weight matrix following global average pooling. For $C, D \in \{E, B\}$, we compute

$$p_{s,CD}(a \rightarrow b) = \frac{\sum_{\ell} \|(J_{s,\ell})_{CD}\|_F^2}{\sum_{\ell} \|J_{s,\ell}\|_F^2}.$$

Thus, squared Frobenius energies are pooled across matrices before normalization, without per-matrix or dimension normalization. Matrices contribute in proportion to their in-frame displacement energy, and off-frame energy is excluded from the denominator. Each panel displays $100 \times \frac{1}{5} \sum_{s=0}^4 p_{s,CD}(a \rightarrow b)$, giving equal weight to the five normalized seed-level shares. Each task interval is processed separately; energies are not pooled across intervals or seeds.

The full model is optimized during training. The checkpoint-displacement summaries analyze only hidden Linear weight matrices, excluding convolutional weights and the final output classifier.

A.7 ViT-Tiny task-100 forward and backward diagnostic

The task-100 forward and backward diagnostic uses seed 0 of the ImageNet-32 class-pair stream. We load the continual ViT-Tiny checkpoint after 100 completed tasks and evaluate all 1,200 training images from the 100th binary task, corresponding to ImageNet classes 304 and 783. Examples are processed without shuffling in batches of 100. Because each image produces 64 patch tokens and one class token, each analyzed layer contributes

$$N = 1,200 \times 65 = 78,000$$

image-token sites.

We analyze the MLP up-projections in Transformer blocks 1, 6, and 12, each with

$$W_{\text{up}} \in \mathbb{R}^{768 \times 192}.$$

The up-projection family was selected before examining the input- and error-side statistics because its fitted Marchenko–Pastur supports were more consistent across depth than those of the corresponding down-projections. For each selected matrix, we freeze the checkpoint-local compact singular value decomposition

$$W_{\text{up}} = U \Sigma V^\top.$$

Let $h_i \in \mathbb{R}^{192}$ denote the activation entering W_{up} at image-token site i . We define

$$\bar{h} = \frac{1}{N} \sum_{i=1}^N h_i, \quad C_h = \frac{1}{N} \sum_{i=1}^N (h_i - \bar{h})(h_i - \bar{h})^\top.$$

For singular direction k , the input-covariance energy is

$$E_k^V = v_k^\top C_h v_k = \text{Var}(h^\top v_k).$$

This measures the activation variance directly along the right singular direction v_k .

Let $y_i = W_{\text{up}} h_i$ denote the projection output and

$$\delta_i = \frac{\partial L}{\partial y_i} \in \mathbb{R}^{768}$$

the backpropagated error at that output. We define the uncentered error second-moment matrix

$$M_\delta = \frac{1}{N} \sum_{i=1}^N \delta_i \delta_i^\top.$$

The corresponding backpropagated-error second moment along the left singular direction u_k is

$$E_k^U = u_k^\top M_\delta u_k = \mathbb{E}[(\delta^\top u_k)^2] .$$

The backward pass uses cross-entropy summed over each minibatch and divided by the complete task size, so the absolute scale of E_k^U inherits this dataset-average loss normalization.

Within each layer, we compare the upper edge E^+ with the MP bulk B separately for the two quantities:

$$R_V = \frac{\text{median}_{k \in E^+} E_k^V}{\text{median}_{k \in B} E_k^V}, \quad R_U = \frac{\text{median}_{k \in E^+} E_k^U}{\text{median}_{k \in B} E_k^U}.$$

These ratios are computed using all directions assigned to the corresponding spectral regions in the complete spectrum. The range $0 \leq \sigma \leq 5$ used in the figure is only a display limit; directions with $\sigma > 5$ remain included in the reported statistics. No lower-edge outliers are detected in the three selected matrices.

Staats et al. [14] measure the maximum overlap between each right singular vector v_k and the eigenvectors of C_h . Here, instead, we measure covariance energy directly along each weight singular direction and additionally measure the returning error signal along the corresponding left singular direction. The diagnostic therefore tests whether the same non-random spectral region is exposed to both forward activations and backward errors. It is a single-checkpoint, single-seed observational analysis and is not interpreted as standalone evidence that the upper edge causes loss of plasticity.

B Additional Results

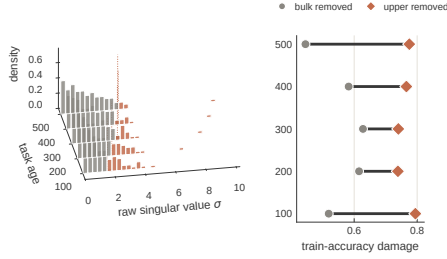


Figure 6: Training produces upper edges beyond the random bulk. Left: MLP spectra across tasks. Right: training accuracy decrease after removing matched numbers of upper-edge or bulk singular components.

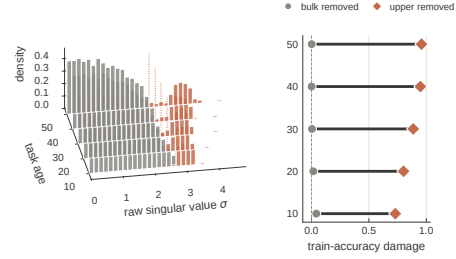


Figure 7: Training produces upper edges beyond the random bulk. Left: ResNet-18 spectra across tasks. Right: training accuracy decrease after removing matched numbers of upper-edge or bulk singular components.

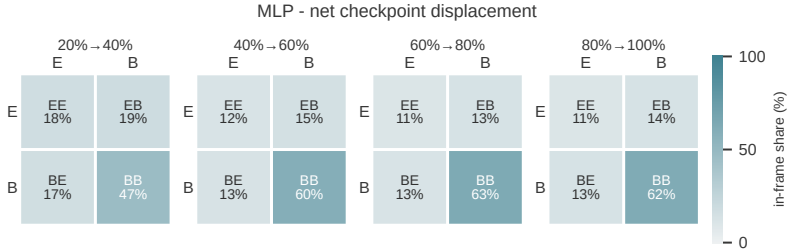


Figure 8: MLP checkpoint motion shifts toward the bulk rather than toward existing spectral-edge directions. Each panel reports the in-frame share of the net checkpoint displacement expressed in the singular basis frozen at the beginning of the corresponding task interval. From the earliest to the latest interval, the combined edge-touching share ($EE + EB + BE$) decreases from approximately 54% to 38%, while the BB share increases from 47% to 62%. This trend differs from the increasing edge concentration observed for ResNet-18 and ViT-Tiny. We hypothesize that the difference may arise from the simpler two-hidden-layer MLP architecture and its use of zero-momentum SGD rather than Adam. Because architecture and optimizer vary together in these experiments, this diagnostic does not isolate which factor produces the difference.