Weights describe, representations predict: a cross-architecture topological study of learning and generalisation

Anonymous authors
Paper under double-blind review

Abstract

How a neural network stores what it learns in its weights remains poorly understood. Building on topological studies of weight space, we read each weight tensor at each training checkpoint as a point cloud and ask two questions: does the *shape* of the weights reorganise during training in predictable, architecture-specific ways, and does that shape *predict* how well the network generalises? We probe both with one instrument, a unified five-category topological screen (Mapper graphs, persistent homology, network-science descriptors, discrete Ricci curvatures, and intrinsic dimension), applied to every layer of five architecture families (MLP, plain CNN, ResNet, LSTM, ViT) over three seeds and tracked across training. Our central finding is a dissociation: weight shape *describes* how learning organises but does not *predict* generalisation; the held-out representations do.

The shape does reorganise, with a different organising principle in each family. Convolutional networks develop a per-filter topological *sweet spot*, a single layer whose dominant H_1 cycle sharpens during training and whose position shifts systematically with architecture, from the last convolution in a plain LeNet to the first 3×3 layer after the stem in a ResNet. Other families organise analogously, and beneath all of them the intrinsic dimension of the weight clouds contracts during training.

Weight shape does not, however, predict generalisation. A strong apparent link to the generalisation gap ($\rho = 0.92$) collapses ($\rho = 0.10$) once the degree of fitting is equalised: like the loss, the weights record how far the network has fit its training data. The predictive signal appears instead in the representations induced on *held-out* inputs: the same screen, applied there, does predict the gap, label-free and under extrapolation across datasets and architectures.
All code, experiments and figures are released as an anonymised public repository (released upon acceptance)https://anonymous.4open.science/r/TDA-NN-243F/README.md.

1 Introduction

A trained neural network stores everything it has learned in its weights, yet our vocabulary for describing *how* that knowledge is organised geometrically remains thin. The gap matters for interpretability, where the arrangement of weights is the substrate on which mechanistic explanations must ultimately rest. It matters for generalisation theory, where the effective complexity of a trained network is increasingly understood to be far lower than its parameter count suggests (Zhang et al., 2017; Neyshabur et al., 2017): low intrinsic dimensionality recurs as an explanation both for data (Pope et al., 2021) and for the parameters that fit it (Li et al., 2018a; Aghajanyan et al., 2021), and a geometric account of *why* would be valuable. And it matters for architecture design, since knowing which layers or components actually organise the learned structure could inform where capacity is best spent. A description of weight organisation that is comparable *across* architectures would speak to all of these, but no such description exists.

A productive line of work treats the weights of a layer as a point cloud and studies its *shape* with topological data analysis (TDA). Gabella (2021) represented each neuron of an MLP by its incoming weight vector and showed, via the Mapper algorithm, that during training these points trace branching trees (under

symmetric initialisation) or smooth two-dimensional "learning surfaces" (under small random initialisation). Brüel Gabrielsson & Carlsson (2018) represented each 3×3 convolutional filter as a point in $\mathbb{R}^9$ and found that the filters of well-trained CNNs concentrate on the same Klein-bottle-shaped manifold earlier found in the space of natural image patches (Carlsson et al., 2008), with its prominence correlating with generalisation. A parallel and largely disjoint line studies *intrinsic dimension*: Ansuini et al. (Ansuini et al., 2019) report a characteristic "hunchback" profile of the intrinsic dimension of *activations* with network depth, while Birdal et al. (Birdal et al., 2021) bound the generalisation error by the fractal (persistent-homology) dimension of the optimisation *trajectory* in weight space.

These results are suggestive, but they remain fragmented in ways that make it hard to tell whether they are isolated curiosities or facets of a common principle. Each uses a *single descriptor* (a Mapper graph, or one persistence statistic, or one dimension estimator), so it is unknown whether the phenomena co-occur or are visible only through one lens. Each also studies a *different object*, namely the weights, the activations, or the trajectory, and these are rarely measured on the same networks, so cross-paper comparisons mix confounds. And each is demonstrated on essentially *one architecture family* (MLP, or CNN, or a transformer encoder), leaving open the first question of this paper (**Q1**): does weight-space topology reorganise in *predictable* ways as the defining ingredients of modern architectures (normalisation, residual connections, gating, attention) are introduced? And because the prior weight-space studies relate topology to generalisation without controlling how much of the training data each network has fit, conflating interpolation across differently-fit networks with genuine extrapolation to held-out error, a second question follows (**Q2**): if the shape of the weights does organise, does it *predict* anything about performance on held-out data, or does it merely record the training process? The two questions structure the paper, and the title states its answers.

We answer this by applying a single, uniform *five-category screen* to architecture-specific weight point clouds, tracked across training, for five families (MLP, plain CNN, ResNet, LSTM, ViT), each repeated over three seeds (Sec. 3). The screen comprises five categories: Mapper graphs under two complementary lenses (an L_2 norm and a PCA-3 projection), persistent homology of the weight cloud, network-science descriptors of the Mapper graph, two discrete Ricci curvatures of that graph, and intrinsic dimension measured both spatially (TwoNN) and temporally (Birdal's PHD). Because every architecture is screened with the identical pipeline and the same scale-invariant summaries, the descriptors are directly comparable across families even though the underlying tasks differ. Throughout, we separate *controlled* comparisons (dataset fixed, one architectural variable changed) from unavoidably qualitative contrasts across objects or datasets (Sec. 3.5).

We introduce the unified screen and release it, with all experiments, as an anonymised public repository (released upon acceptance). The results are organised by the two questions: five structural observations (O1–O5) answer Q1, describing *how* and *where* weight geometry organises and how that location depends on the architecture; two observations on generalisation (O6–O7) answer Q2, locating which object carries the predictive signal.

Structure (Q1: O1–O5). In CNNs a per-filter topological *sweet spot* exists, a single convolutional layer whose dominant H_1 cycle grows sharply over training, and its position shifts with the architecture: it sits at the last convolution in a plain shallow CNN and at the first 3×3 layer after the stem in a ResNet (**O1**). The cleanest leg of this comparison, a plain CNN versus a ResNet *both* on CIFAR-10, isolates the batch-normalisation-plus-skip variable, and a two-leg ablation decomposes it (Sec. 5.4): skip connections alone leave the sweet spot in place, batch normalisation alone deepens the organisation, so the relocation is emergent from their combination. A depth-10 factorial then adds a regime caveat: the sweet spot exists only in the high-learning-rate (feature-learning) regime, vanishing at low learning rate even in CNN-4 while fitting improves, and part of batch normalisation's role is to make that regime accessible at depth. The same layer is flagged independently by a positive shift in the Forman–Ricci curvature of its Mapper graph (**O5**). LSTMs relocate the phenomenon from a layer to a gate *role*, the output gate of the deepest layer's input-to-hidden weight (**O2**). Our ViT-mini compresses feed-forward weight dimension increasingly with depth, a profile we contrast qualitatively with Ansuini et al.'s activation hunchback (**O3**). And beneath all of these, the spatial intrinsic dimension of the weight clouds contracts during training in every family, while computing Birdal's trajectory dimension on the *same* runs shows the optimisation path is equally low-dimensional (**O4**).

Generalisation (Q2: O6–O7). Whether any of this topology *predicts* generalisation turns out to depend on the object it is read from. Under a controlled label-noise intervention, in which every run reaches the same train accuracy, the weight descriptors do not predict the gap (**O6**): a strong apparent link ($\rho = 0.92$) proves an artefact of unequal fitting, collapsing to $\rho = 0.10$ once fitting is equalised. But the *same screen*, applied to the representations the network induces on *held-out* inputs, an object the training loss does not determine, does predict it, label-free and under extrapolation across datasets and architectures (**O7**). This is the dichotomy of the title: the weights *describe* the training process; the held-out representations *predict* the outcome on new data.

2 Related work

Topology of network weights. The work closest to ours studies the weights directly. Gabella (2021) built Mapper graphs of the incoming-weight clouds of a one-hidden-layer MLP on MNIST, using the L_2 norm and PCA lenses to expose, respectively, branching trees and 2D learning surfaces, but considered only that single architecture and only the Mapper descriptor. Brüel Gabrielsson & Carlsson (2018); Carlsson & Brüel Gabrielsson (2023) analysed the persistent homology of 3×3 filter clouds across many CNNs trained on MNIST, CIFAR and SVHN, establishing the recurring Klein-bottle loop and relating its presence to generalisation, though across networks whose degree of fitting was not held fixed (the confound O6 isolates); their question is *which* topological feature recurs across networks, not *where* within a given network the organisation is strongest, nor how that location depends on architecture. Pérez-Fernández & Naitzat (2021) use persistent homology of weights as a *similarity metric* between trained models, a different goal. Relative to all three, we advance in three respects: we screen the weights with five descriptor families rather than one, we track them across training rather than at convergence only, and we identify the architecture-dependent *position* of the topological sweet spot (O1) and corroborate it with curvature (O5) and the L_2/PCA-3 surface dichotomy extended to convolutional filters (Sec. 4). None of these is reported in the prior weight-space work.

Topology of activations. A larger and largely separate literature applies TDA to *activations* rather than weights. Naitzat et al. (2020) show, on binary-classification benchmarks, that well-trained networks progressively simplify the topology of the input manifold with depth; Ballester et al. (2022) use activation persistence to compare layer representations, and Kushnareva et al. (2024) apply zigzag persistence to large-language-model activations for layer pruning. These works share our toolkit but not our primary object: activations describe how data is transformed, whereas weights describe what the network has become. Our structural findings (O1–O5) keep to the weights, so that a single descriptor is defined identically for a convolution, a recurrent gate and an attention head; O7 then turns the same screen on *held-out* activations, precisely because the representation, unlike the weights, is not determined by the training loss.

Predicting generalisation from unlabelled data. A separate line estimates a model's test performance without test labels, from statistics of its behaviour on unlabelled held-out inputs: the confidence-threshold estimator of Garg et al. (2022), the feature-distribution regression of Deng & Zheng (2021b), prediction-confidence shifts (Guillory et al., 2021), the agreement of two independently trained models (Jiang et al., 2022), and the in- versus out-of-distribution accuracy line of Miller et al. (2021). A complementary strand predicts accuracy from the weights (Unterthiner et al., 2020) or from complexity measures (Jiang et al., 2020; Neyshabur et al., 2017). Closest to us, Corneanu et al. (2020) predict the train–test gap from the persistent topology of activation correlations, with no test set. O7 sits in this space, and we are explicit that predicting the gap from the geometry of held-out representations is, on its own, an incremental step beyond this line: like Corneanu et al. we read a topological descriptor of the held-out activations, here its persistent homology and intrinsic dimension under leave-one-noise-level-out *extrapolation* across datasets and architectures. Our contribution is not that the held-out representation predicts, which these works already suggest, but the controlled *dissociation* that brackets it: the *same* five-category screen, applied to the weights of the *same* networks under a regime that equalises fitting, does *not* predict the gap (O6), whereas applied to the held-out representations it does (O7). The novelty is therefore the object contrast at fixed method and fixed runs, which isolates *where* the generalisation signal lives, rather than a new unlabelled estimator; none of the works above measures both objects with one instrument or reports the negative half.

Intrinsic dimension. The intrinsic dimension of deep networks has been studied on three different objects. Ansuini et al. (2019) measure the TwoNN dimension of *activations* of AlexNet, VGG and ResNet on ImageNet and find a non-monotone "hunchback" across depth whose final value predicts accuracy. Birdal et al. (2021) introduce the persistent-homology dimension of the optimisation *trajectory* and prove a generalisation bound in terms of it. Bushuiev et al. (2024) track the dimension of transformer token *embeddings* through training, reporting an expansion-then-compression dynamic. Our TwoNN measures the *spatial* dimension of the weight clouds at each checkpoint, a fourth object, complementary to all three. We make the relationship to Birdal explicit and *controlled*: we re-implement their trajectory PHD and compute it on the very same runs (Sec. 8, Table 5), so the spatial-vs-temporal contrast is not confounded by differing models or data. The contrast with Ansuini, by construction, cannot be controlled in this way, since their object (activations), architectures and dataset (ImageNet) all differ from ours, so we present it as a qualitative juxtaposition, not a replication. We also distinguish our quantity from Xue et al. (2022), who use "intrinsic dimension" to name a pruning search.

Discrete curvature in deep learning. Discrete Ricci curvature has entered deep learning chiefly through graph neural networks, where it is computed on the *message-passing graph*: Topping et al. (2022) relate negative curvature to over-squashing, and Coupette et al. (2023) systematically compare Ollivier-curvature variants on graphs. We instead compute Forman and Ollivier curvature on the *Mapper graph of a weight cloud*, a use we are not aware of in prior work, and find that it independently flags the same CNN sweet-spot layer identified by persistence (O5), providing a second, methodologically distinct line of evidence for our main claim.

Loss-landscape and weight-space geometry. A large literature characterises the geometry of weight space through the loss surface rather than the weights as a point cloud. Mode-connectivity results show that independently trained solutions are joined by low-loss paths (Garipov et al., 2018; Draxler et al., 2018), and linear mode connectivity relates this structure to training stability (Frankle et al., 2020; Frankle & Carbin, 2019). A parallel strand ties the *flatness* of a minimum to generalisation, from sharp-minima diagnostics (Keskar et al., 2017) and PAC-Bayes bounds (Dziugaite & Roy, 2017) to sharpness-aware optimisation (Foret et al., 2021) and large-scale empirical surveys (Jiang et al., 2020). These descriptors read the loss landscape *around* a point in weight space; ours read the intrinsic shape of the weight tensors *at* that point, tracked across training and defined identically for convolutions, gates and attention heads. The two views are complementary, and our O6 result is consistent with the flatness line in spirit: like sharpness under leave-one-noise-level-out evaluation (Sec. 10), our weight descriptors do not predict the gap once fitting is equalised.

Representation geometry. The geometry of *representations*, the object of our O7, has its own non-topological literature. Neural collapse describes how last-layer features and classifiers converge to a highly symmetric configuration in the terminal phase of training (Papyan et al., 2020), and representation-similarity measures such as CKA (Kornblith et al., 2019) and SVCCA (Raghu et al., 2017) quantify how representations differ across layers, models and training runs. These tools describe the structure of held-out representations with linear-algebraic summaries; O7 instead reads the same representations with the topological screen used on the weights, so that the weights-versus-representations dissociation (O6 versus O7) is measured with one instrument. We use these representation-geometry baselines as points of comparison rather than as the primary descriptor.

3 Methods

3.1 Architecture-specific point-cloud mappings

The object we study is not the scalar loss or the network's activations but the *weights themselves*, viewed as geometry. At every training checkpoint $t = 0, 1, \ldots, T$ we take a learnable weight tensor W_t and turn it into a finite point cloud $X_t = \phi(W_t) \subset \mathbb{R}^d$ through an architecture-specific embedding ϕ. Because every topological and geometric descriptor in the remainder of this section is a function of X_t alone, tracking those descriptors along the trajectory $t = 0, 1, \ldots, T$ amounts to watching the *shape* of the weights evolve as the

network learns. The only design choice the embedding involves is what counts as a "point", and the natural answer differs from one architecture to the next; we therefore use a small family of mappings, each chosen so that a point corresponds to a coherent, repeated computational unit of the layer.

For a fully connected layer the unit is a neuron: a dense weight matrix $W \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$ contributes one point per row, so that $X_t \subset \mathbb{R}^{n_{\text{in}}}$ is the cloud of incoming weight vectors of its neurons. Convolutional layers admit two complementary readings. In the *per-filter* reading, which follows Brüel Gabrielsson & Carlsson (2018), each individual 3×3 kernel $W_{c_{\text{out}}, c_{\text{in}}, :, :}$ is flattened to a single point in $\mathbb{R}^9$, giving a dense cloud whose Klein-bottle-type structure is the central object of our CNN findings; in the coarser *per-channel* reading each output channel is vectorised as one point in $\mathbb{R}^{c_{\text{in}} \cdot 3 \cdot 3}$, which trades resolution for a global view of how channels are arranged. For an LSTM the meaningful unit is a gate. PyTorch stores each layer's recurrent weights in two packed matrices: the input-to-hidden W_{ih}, which maps the current input x_t, and the hidden-to-hidden W_{hh}, which maps the previous hidden state h_{t-1} (the recurrent connection proper). Each matrix stacks the four gates (input i, forget f, cell g, output o) along its rows, so we split it by kind $\in \{\text{ih}, \text{hh}\}$ and by gate $\in \{i, f, g, o\}$, and the rows of each gate then form a separate cloud in $\mathbb{R}^{n_{\text{in}}}$, so that we can identify the *functional role*, and not just the layer, in which topological structure concentrates.

Transformers offer the richest choice, and we keep three mappings at different granularities. The coarsest, *attn-per-head-vec*, summarises a whole block as a single cloud of n_{heads} points, each head vectorised into $\mathbb{R}^{3 d_{\text{head}} d_{\text{model}}}$; the finer *attn-per-head-rows* instead resolves a single head into its $3 d_{\text{model}}$ row vectors in $\mathbb{R}^{d_{\text{head}}}$, exposing structure *within* a head; and *ffn-rows* takes the rows of each block's feed-forward matrix W_1 as n_{ffn} points in $\mathbb{R}^{d_{\text{model}}}$. The last mapping is deliberately chosen to be directly comparable to the MLP construction, so that a feed-forward block of a Transformer and a plain dense layer are screened on exactly the same footing.

3.2 Persistent homology of the weight cloud

We summarise only the constructions we use; see Edelsbrunner & Harer (2010), Carlsson (2009) and Ghrist (2008) for thorough treatments. Given $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ we build the Vietoris–Rips filtration: for a scale $\epsilon \geq 0$ the simplicial complex $\text{VR}_\epsilon(X)$ contains a k-simplex $\{x_{i_0}, \ldots, x_{i_k}\}$ whenever $\|x_{i_a} - x_{i_b}\| \leq \epsilon$ for all a, b. As ϵ increases from 0 to ∞ the complexes are nested, $\text{VR}_\epsilon(X) \subseteq \text{VR}_{\epsilon'}(X)$ for $\epsilon \leq \epsilon'$, and applying homology $H_k(\cdot; \mathbb{F}_2)$ yields a sequence of vector spaces connected by linear maps, a *persistence module*. Its decomposition produces a multiset of intervals $\text{Dgm}_k(X) = \{(b_i, d_i)\}_i$, the degree-$k$ persistence diagram, where a feature born at scale b_i and dying at d_i has *lifetime* $\ell_i = d_i - b_i$. The diagram is stable: small perturbations of X move it by a bounded amount in bottleneck distance (Cohen-Steiner et al., 2007), which underwrites the use of its summaries as descriptors. We use H_0 (connected components) and H_1 (independent loops), computed with Ripser, subsampling to $n \leq 1500$ points when larger.

From Dgm_k we derive scale-aware summaries. Writing the finite lifetimes in decreasing order $\ell_{(1)} \geq \ell_{(2)} \geq \ldots$, we use the total persistence $L_k = \sum_i \ell_i$, the persistence entropy (Atienza et al., 2020)

$$\text{PE}_k(X) \;=\; -\sum_i p_i \log p_i, \qquad p_i = \frac{\ell_i}{L_k}, \tag{1}$$

which is maximal ($\log |\text{Dgm}_k|$) when all features are equally long-lived (noise) and drops when one cycle dominates, and the *signal-to-noise ratio*

$$\text{SNR}_k(X) \;=\; \frac{\ell_{(1)}}{\text{median}_i\, \ell_i}, \tag{2}$$

a scale-invariant contrast between the dominant cycle and the bulk, in the spirit of the standard barcode heuristic that long bars are signal and short bars are noise (Ghrist, 2008; Cohen-Steiner et al., 2007).

Headline descriptor. Our headline CNN/LSTM descriptor, used throughout O1, O2 and O6, is SNR_1, the *Klein signal-to-noise*: a $\Delta\text{SNR}_1 \gg 0$ over training means a single H_1 loop has emerged from the noise floor: the weights have organised onto a manifold with a dominant cycle, the Klein-bottle-type structure of Brüel Gabrielsson & Carlsson (2018). This is the single quantity the reader needs to follow the structural

findings: wherever we speak of a layer's or gate's *sweet spot*, we mean the locus of the largest $\Delta\mathrm{SNR}_1$. In plain terms, the sweet spot is the one layer (or gate) that, as the network learns, most sharply develops a single dominant loop in its weight cloud while the others stay topologically noisy: it is where topological organisation concentrates. We also record Betti numbers $\beta_k(\epsilon) = \dim H_k(\mathrm{VR}_\epsilon)$ at the quartiles of the birth distribution, and the stable vectorisation of Dgm_1 as a persistence image (Adams et al., 2017) (an alternative stable summary to the persistence landscape (Bubenik, 2015)).

3.3 Mapper graph and its network/curvature descriptors

The Mapper construction (Singh et al., 2007; Lum et al., 2013) summarises X as a graph. Given a lens $f : X \to \mathbb{R}^m$, cover $\mathbb{R}^m$ by overlapping hypercubes $\{U_j\}$; cluster the preimage $f^{-1}(U_j)$ within each cube; create one node per cluster and connect two nodes whose clusters share a point. We use two lenses, the L_2 norm $f(x) = \|x\|_2$ (Gabella's default, which exposes radial branching) and the first three PCA coordinates (which expose 2D learning surfaces), with 15 cover intervals, 40% overlap, and DBSCAN clustering ($\epsilon = 0.5$, minPts $= 5$). On the resulting graph $G = (V, E)$ we compute degree statistics, betweenness centrality, the Louvain modularity Q, the cycle rank $\beta_1(G) = |E| - |V| + c$ (c = number of components), and the diameter of the largest component. The full screen records all of these for every layer and checkpoint (released in the per-run files), but only two of them carry findings in this paper: the cycle rank $\beta_1(G)$, which separates tree-like from surface-like Mapper graphs (Sec. 4), and the curvatures below (O5); the remaining graph statistics are reported for completeness. Table 3 summarises how every descriptor of the screen is read and where it is used.

We further equip G with two notions of discrete Ricci curvature. For the **Forman–Ricci** curvature of an edge $e = (u, v)$ in an unweighted graph we use the triangle-augmented form of Samal et al. (2018),

$$\mathrm{Ric}_F(e) \;=\; 4 - \deg(u) - \deg(v) + 3\,\big|\{\text{triangles on } e\}\big|, \tag{3}$$

that is, the bare graph (1-complex) curvature $4 - \deg(u) - \deg(v)$ of Sreejith et al. (2016) augmented by the contribution of the 2-cells (triangles) on e; it is positive on tightly knit (sphere-like) neighbourhoods and negative on tree-like (hyperbolic, branching) ones. The **Ollivier–Ricci** curvature (Ollivier, 2009; Lin et al., 2011) compares neighbourhood mass transport,

$$\mathrm{Ric}_O(e) \;=\; 1 - \frac{W_1(\mu_u, \mu_v)}{d(u, v)}, \tag{4}$$

where μ_x is the (lazy) uniform measure on the neighbours of x, W_1 is the Wasserstein-1 distance and d the graph distance. Both are summarised by their mean/std/min/max over E. (For the multi-seed sweep we report Ric_F, which is $O(|E|)$; Ric_O requires an optimal-transport solve per edge and is computed for a single representative run (seed 42) only.)

3.4 Intrinsic dimension: spatial (TwoNN) and temporal (PHD)

The **TwoNN** estimator (Facco et al., 2017), a development of the maximum-likelihood approach of Levina & Bickel (2004), measures the local intrinsic dimension (ID) of X at a fixed checkpoint. For each point let $r_1 \le r_2$ be the distances to its first two neighbours and $\rho = r_2/r_1$. Under a locally uniform density of intrinsic dimension d, ρ follows a Pareto law with $\Pr(\rho > s) = s^{-d}$, so d is the slope of the empirical $\big(\log\rho,\ -\log(1 - F(\rho))\big)$ relation,

$$\hat{d}_{2\mathrm{NN}} \;=\; \frac{\sum_i \log\rho_i \left(-\log[1 - F(\rho_i)]\right)}{\sum_i (\log\rho_i)^2} \quad \text{(linear fit through the origin of } -\log(1 - F) \text{ vs. } \log\rho). \tag{5}$$

In plain terms, TwoNN estimates how many independent directions the points really vary along, so a large drop over training means a scattered, high-dimensional weight cloud has condensed onto a low-dimensional manifold. We complement it with PCA-99%/95% (number of principal components retaining that fraction of variance) as classical baselines.

Family	Model (key traits)	Dataset (task)	Supports
MLP	one hidden layer	MNIST (10-way)	Gabella control
plain CNN	LeNet-3×3 (3 conv, no BN/skip)	MNIST (10-way)	O1 (trend)
plain CNN	CNN-4 (4 conv, no BN/skip, wider)	CIFAR-10 (10-way)	O1, O6, O7
CNN ablation	CNN-4+BN (4 conv, BN, no skip)	CIFAR-10 (10-way)	O1 ablation
CNN ablation	CNN-4+skip (4 conv, skip, no BN)	CIFAR-10 (10-way)	O1 ablation
CNN factorial	CNN-10 2×2 (plain / +BN / +skip / +BN+skip)	CIFAR-10 (10-way)	O1 regime study
ResNet	ResNet-18 (17 conv, BN + skip)	CIFAR-10 (10-way)	O1, O7
ResNet	ResNet-34 (33 conv, BN + skip)	CIFAR-10 (10-way)	O1/O4 scaling
LSTM	two-layer LSTM	IMDB (binary)	O2
ViT	ViT-mini (6 blocks, 4 heads)	CIFAR-10 (10-way)	O3

Table 1: The thirteen trained models (five families), each over three seeds $\{7, 42, 123\}$ (the ResNet-34 scaling run over five). The last column lists the family-specific observations each model anchors; the spatial/temporal intrinsic-dimension contraction (O4) is measured on all six representative models (Table 4), and the curvature corroboration (O5) on the CNNs. The plain CNN-4 versus residual ResNet-18 pair, both on CIFAR-10, is the controlled comparison that isolates batch normalisation and skip connections (O1). The two ablation variants decompose that compound variable one ingredient at a time, and the depth-10 factorial decomposes it at depth, where it exposes the training-regime dependence of the sweet spot (Sec. 5.4); the factorial cells use the one recipe that trains them all (He init, lr 0.01), with the BN-bearing cells retrained at lr 0.05 for the regime test. The generalisation studies (O6, O7) add controlled label-noise, train-size, and CIFAR-100 sweeps described in their sections.

Whereas TwoNN sees the *spatial* cloud at one checkpoint, the **persistent-homology dimension** (PHD) of Birdal et al. (2021) measures the fractal dimension of the optimisation *trajectory* $\{\theta_0, \ldots, \theta_T\}$ in parameter space. With $E_\alpha^0(A) = \sum_{(b,d)\in \mathrm{Dgm}_0(A)} (d-b)^\alpha$ the α-weighted total H_0 persistence of a finite set A, the PHD is the exponent $\dim_{\mathrm{PH}}^\alpha$ such that

$$E_0^\alpha(A_n) \;\sim\; C\, n^{(\dim_{\mathrm{PH}}^\alpha - \alpha)/\dim_{\mathrm{PH}}^\alpha} \tag{6}$$

for subsamples $A_n \subset \{\theta_t\}$ of size n; we recover it (with $\alpha = 1$, so E_0^1 is the minimum-spanning-tree length) by fitting the slope β of $\log E_0^1(A_n)$ versus $\log n$ and inverting, $\dim_{\mathrm{PH}} = \alpha/(1-\beta)$. The two dimensions are complementary: TwoNN asks how low-dimensional the weights are *at* a checkpoint, PHD asks how low-dimensional the path *between* checkpoints is. In practice we read them as follows. A large negative ΔID (init→end change of the TwoNN estimate) means the weight cloud has condensed onto a manifold of far lower dimension than its ambient space, i.e. the layer's units have become strongly co-organised. A small PHD (1–3, against millions of parameters) means the entire optimisation path explored only a low-dimensional fractal subset of parameter space. When both are small, as we find in every model we train (Sec. 8), the effective complexity of training is far below the parameter count, spatially and temporally at once.

3.5 Experimental design and the validity of comparisons

We train thirteen models spanning five architecture families (Table 1), each with three random seeds $\{7, 42, 123\}$ (the ResNet-34 scaling run of Sec. 5.5 adds two more, $\{1, 2\}$, for five); every reported delta is a mean $\pm$ standard deviation over those seeds, computed between the random-initialisation checkpoint and convergence. The screen, its hyper-parameters, and all scale-invariant summaries (Eqs. 1–5) are identical across families. Because three seeds are few, we do not lean on point estimates for the headline claims: the sweet-spot localisation (O1) is reported as a per-seed argmax that is identical across seeds and datasets, and the generalisation correlations (O6) are reported with bootstrap confidence intervals and permutation tests rather than a single ρ (the controlled-regime interval straddles zero; the negative result does not depend on a particular seed).

Because no single dataset is natural for all of these architectures, our comparisons are of three kinds (Table 2), and we are explicit about which evidence each finding rests on. This design choice is justified empirically in Sec. D: holding the architecture fixed (CNN-4) while sweeping four datasets that span a tenfold range of class count shows that the sweet-spot position is driven by the architecture, not by task complexity, so the necessar-

Kind	Setup	What we compare	Example findings (evidence)
(i) Controlled, within-paper	dataset fixed; one architectural variable changed	descriptors head-to-head	O1: CNN-4 vs ResNet-18, both CIFAR-10, isolates BN+skip; O4: TwoNN vs PHD on the same runs. *Strongest.*
(ii) Relative, scale-free	datasets differ across families	only within-run init→end *changes* of scale-invariant descriptors	O2, O3 across CNN/LSTM/ViT; the LeNet/MNIST point in O1. *Trend-level.*
(iii) Qualitative, cross-paper	object, architecture and dataset all differ	direction of an effect only, no shared metric	O3 vs Ansuini's activation hunchback. *Juxtaposition.*

Table 2: The three kinds of comparison the paper rests on, from strongest to weakest. Only kind (i) isolates a single cause; kinds (ii) and (iii) support trends and motivate, but do not settle, the claims. Each finding's category is flagged in the results.

ily different datasets across families do not by themselves drive the cross-family conclusions. **(i) Controlled within-paper comparisons**, in which a single architectural variable is changed with the dataset held fixed. The strongest leg of the sweet-spot result (O1) is of this kind: CNN-4 and ResNet-18 are both trained on CIFAR-10 for the same 10-way task, so the shift of the sweet spot from an intermediate convolution to the first post-stem layer isolates the addition of batch normalisation and skip connections. Likewise, the spatial-versus-temporal intrinsic-dimension contrast (O4) is fully controlled: TwoNN and Birdal's PHD are computed on the *same* checkpoints of the *same* runs. **(ii) Relative, scale-free cross-architecture comparisons**. When we compare a CNN to an LSTM to a ViT, the datasets necessarily differ (CIFAR-10, IMDB, CIFAR-10), so we never compare raw descriptor magnitudes across families; we compare only *within-run* init→end *changes* of scale-invariant quantities (signal-to-noise, persistence entropy, normalised dimension), which removes the dependence on ambient dimension and dataset scale. The LeNet/MNIST datapoint in O1 is of this relative kind: it supports the trend but, because it changes both dataset and architecture relative to the CNN-4/ResNet pair, we do not treat it as part of the controlled BN+skip comparison. **(iii) Qualitative cross-paper contrasts**, chiefly with Ansuini et al.'s activation-dimension hunchback (O3): the object (activations vs weights), the architectures, and the dataset (ImageNet vs CIFAR-10) all differ, so we present the contrast as a juxtaposition that motivates future controlled study, not as a head-to-head test. Throughout the results we flag each finding's category so that the reader can weight the evidence accordingly.

3.6 The screen at a glance

Table 3 collects every descriptor of the five-category screen (Secs. 3.2–3.4), what it measures on the weight point cloud, how a training-time change is interpreted, and which observations rest on it. It is the reading key for all results sections.

Robustness of the Mapper-dependent descriptors. Our lead findings do not use the Mapper graph at all: O1–O4 are computed from Vietoris–Rips persistence or nearest-neighbour statistics on the raw point cloud, so they are Mapper-independent by construction. Only the Ricci-curvature corroboration (O5) and the trees-vs-surfaces cycle counts (Sec. 4) read from the Mapper graph. For those quantities, Appendix B reports a sensitivity sweep over the full 27-configuration grid of cover and clustering hyperparameters and a *learned* (autoencoder) lens that replaces the hand-chosen L_2/PCA-3 projections: the trees-vs-surfaces dichotomy is invariant throughout, governed by the resolving dimension of the lens rather than by its choice, while the Forman curvature mean is the one configuration-sensitive descriptor, which is why we present curvature (O5) as corroborative rather than primary evidence.

Descriptor	Object	How a change is read	Used in
SNR_1 (Eq. 2)	H_1 diagram of raw cloud	$\Delta\mathrm{SNR}_1 \gg 0$: one dominant cycle emerges (Klein-bottle-type manifold); defines the *sweet spot*	O1, O2, O6
PE_1 (Eq. 1)	H_1 diagram	drop: lifetime distribution concentrates on few cycles (topological pruning)	O3, O6
Betti numbers, total persistence	H_0/H_1 diagrams	size of the component/loop population at fixed scales	O3, Sec. 4
Mapper cycle rank $\beta_1(G)$	Mapper graph	0: tree-like (radial branching); $\gg 0$: 2-D learning surface	Sec. 4, App. B
Degree, betweenness, modularity Q, diameter	Mapper graph	recorded for completeness; not load-bearing for any finding	released files
Forman / Ollivier curvature (Eqs. 3–4)	Mapper graph edges	positive shift: hub-like condensation onto a well-connected manifold; negative: tree-like branching	O5
TwoNN ID (Eq. 5)	raw cloud at a checkpoint	$\Delta\mathrm{ID} < 0$: spatial compression onto a low-dimensional manifold	O2, O3, O4, O7
Trajectory PHD (Eq. 6)	whole-model checkpoint trajectory	small (1–3): the optimisation path itself is low-dimensional	O4

Table 3: The five-category screen at a glance: each descriptor, the object it is computed on, the interpretation of its training-time change, and the observations that rest on it.

4 Sanity check: reproducing Gabella (2021)

Before extending to modern architectures we verify the pipeline reproduces Gabella's (2021) MLP results. Training the one-hidden-layer MLP on MNIST with all weights initialised to zero, the L_2-lens Mapper graph of the weight cloud traces the expected branching tree, and the persistence screen on this control behaves degenerately by construction: the weights stay pinned near the origin, so the TwoNN intrinsic dimension does not contract ($\Delta\mathrm{ID} = +1.3 \pm 0.2$ on `fc2`, versus large negative drops for every genuinely trained model in Sec. 8) and the Forman curvature of the star-shaped Mapper graph drops maximally ($\Delta\mathrm{Ric}_F = -15.0 \pm 0.0$). With tiny-random initialisation ($\sigma = 10^{-6}$) the first-hidden-layer weights instead spread along a smooth two-dimensional surface, visible under the PCA-3 lens. These controls reproduce Gabella's qualitative picture and confirm that our descriptors react to genuine learning, not to the screen itself; they are not claimed as a contribution.

Trees vs. surfaces extends to convolutional filters. Gabella's distinction between the two Mapper lenses, the L_2 norm (which collapses the radial direction and reveals *branching trees*) and the first PCA coordinates (which reveal *two-dimensional learning surfaces*), was demonstrated only for MLP weights. Running both lenses on the converged 3×3 per-filter clouds (the full screen, Sec. 3.3) shows the same dichotomy holds, and far more strongly, for CNNs: under the L_2 lens every convolutional Mapper graph is a tree (independent-cycle count $\beta_1(G) = 0$ for LeNet, CNN-4 and ResNet-18), whereas under the PCA-3 lens the same clouds become densely connected grids with hundreds to thousands of cycles ($\beta_1(G) = 1634$, 726, 1689 respectively at the sweet-spot layer; Figure 1a). The persistent homology of the raw cloud confirms a rich loop structure invisible to the L_2 Mapper: the H_1 Betti number at the median birth scale is $\beta_1 = 195$ (LeNet conv3), 298 (CNN-4 conv2) and 175 (ResNet layer1.0.conv1; Figure 1b). To our knowledge this is the first demonstration that the Gabella learning-surface picture, not merely the Klein-bottle loop, carries over to convolutional filters.

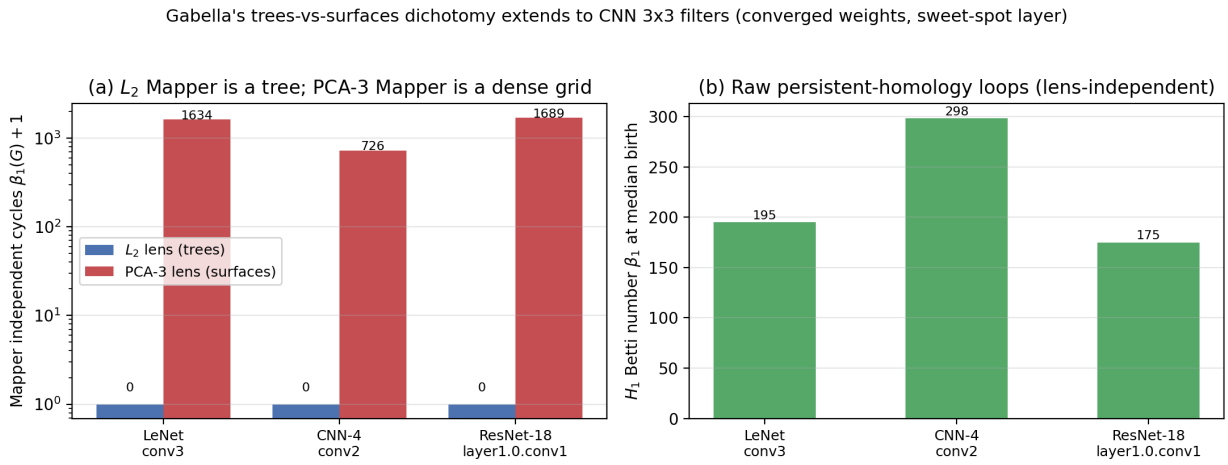

Figure 1: Gabella's two-lens dichotomy on converged 3×3 filter clouds at each CNN's sweet-spot layer. **(a)** Number of independent cycles $\beta_1(G)$ of the Mapper graph (log scale, +1 offset): under the L_2 lens the graph is a *tree* ($\beta_1(G) = 0$), under the PCA-3 lens a dense *grid* ($\beta_1(G)$ in the hundreds to thousands). **(b)** The lens-independent H_1 Betti number of the raw point cloud at the median birth scale, confirming the loop structure is a property of the weights, not of the lens. Values shown for a single representative run (seed 42).

5 Observation 1: the CNN topological sweet spot shifts with normalisation and skip connections

This is our lead structural finding: a topological descriptor of the weights that changes during training and whose *location* in the network is dictated by architectural design choices.

5.1 Experimental setup

We compare three convolutional architectures of increasing inductive bias, all built from 3×3 filters: LeNet-3×3 (no normalisation, no skips, on MNIST), CNN-4 (deeper and wider, still no normalisation or skips, on CIFAR-10) and a CIFAR-adapted ResNet-18 (batch normalisation and residual connections, on CIFAR-10), each over three seeds (full details in Sec. 3.5). As laid out there, the CNN-4-versus-ResNet leg is a *controlled* comparison (same dataset, same task, with batch normalisation and skip connections the only variables changed), whereas LeNet contributes a supporting datapoint under the relative, scale-free reading. The descriptor of interest is the per-filter H_1 signal-to-noise ratio (Eq. 2); we report its init→end change, ΔSNR, as mean ± standard deviation over seeds.

5.2 Result

In every architecture, the change in ΔSNR identifies a single dominant layer, the topological *sweet spot*, and that layer moves systematically earlier in the network as architectural biases are added (Figure 2). In the plain shallow LeNet the sweet spot is the *last* convolution, conv3 ($\Delta\text{SNR} = +4.53 \pm 0.69$), with the signal accumulating monotonically through depth (conv2 $+3.26 \pm 1.60$, conv1 -0.41 ± 1.40). Widening and deepening the plain network (CNN-4) moves the sweet spot to the *second* convolution, conv2, where the growth is both largest and most pronounced ($+17.6 \pm 7.3$, against $+4.80 \pm 2.14$ at conv3 and $+2.24 \pm 0.87$ at conv4). Adding batch normalisation and skip connections (ResNet-18, on the same CIFAR-10 task as CNN-4) moves it all the way to the *first* 3×3 convolution after the stem, layer1.0.conv1 ($+5.93 \pm 1.52$); the remaining sixteen 3×3 layers all stay below +3.7 and the stem itself barely moves ($+0.55 \pm 0.10$). In each network the sweet-spot estimate lies above zero by more than its seed-to-seed standard deviation and does not overlap the next-largest layer, so the *position* of the sweet spot is a stable, seed-independent property rather than an artefact of any single run. Concretely, conv2 is the argmax-ΔSNR layer in all six plain CNN-4

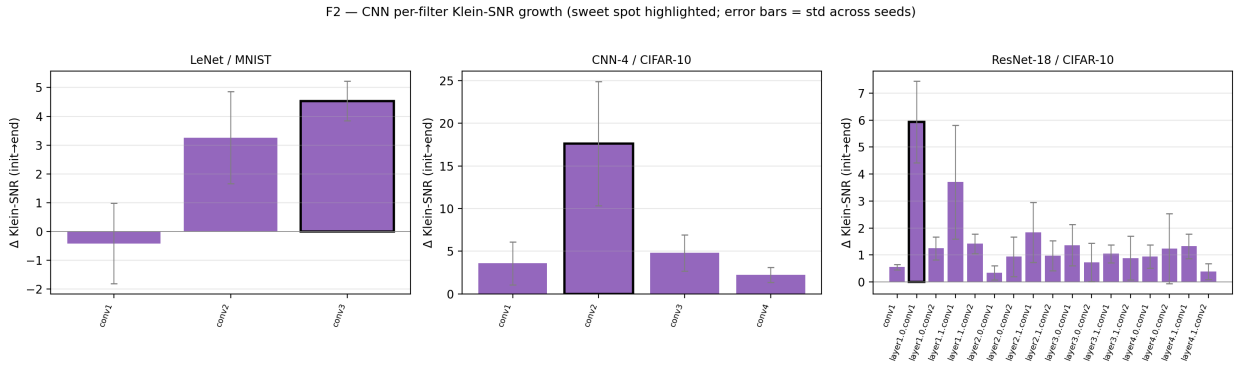

Figure 2: O1: per-filter H_1 signal-to-noise growth (ΔSNR, init→end) for each 3×3 convolution. Bars are means over three seeds; error bars are std; the sweet-spot layer is outlined in black. The sweet spot shifts from the last conv (LeNet) to the second conv (CNN-4) to the first 3×3 after the stem (ResNet-18).

runs (three seeds on CIFAR-10 and three on CIFAR-100); under a null in which the winning layer is uniform over the four convolutions this co-occurrence has $p \approx 0.001$.

5.3 Interpretation

Without normalisation the per-filter point cloud has to "find itself" through the depth of the network: in LeNet the topological organisation accumulates monotonically and only crystallises at the last convolution, while in the wider CNN-4 it concentrates at an intermediate layer once enough filters are present to populate the $\mathbb{R}^9$ manifold.

Batch normalisation (Ioffe & Szegedy, 2015) plus residual connections (He et al., 2016) move the sweet spot all the way to the first 3×3 layer. By smoothing the optimisation landscape (Santurkar et al., 2018a), BN normalises the activation distribution immediately after the stem, so the first residual block can already organise its filters onto the geometric-primitive manifold; the skip path then lets every deeper layer specialise without having to preserve that topology, which is why the 16 deeper ResNet layers show only weak, roughly uniform ΔSNR. The position of the sweet spot is thus a readout of how early the architecture lets geometric primitives form. The ablation of Sec. 5.4 sharpens this reading: neither ingredient alone reproduces the early relocation. Skip connections by themselves leave the plain profile untouched, batch normalisation by itself strengthens the organisation but spreads it toward the *deeper* layers, and only their combination in the residual architecture moves the locus to the first post-stem layer.

Holding the architecture fixed while varying the dataset confirms that the position is driven by architecture rather than by task complexity: CNN-4's sweet spot stays at `conv2` across CIFAR-10, SVHN and CIFAR-100, a tenfold range of class count, with the *input geometry* (resolution and channel count) the only secondary modulator (the lower-resolution Fashion-MNIST flattens the profile, as it does for LeNet on MNIST). Full per-dataset profiles are in App. D.

5.4 Decomposing the BN+skip variable: neither ingredient alone relocates the sweet spot

The controlled CNN-4-versus-ResNet-18 leg of O1 isolates batch normalisation and skip connections *jointly*: the two ingredients are added together (along with depth), so the comparison cannot say which of them drives the relocation of the sweet spot. To decompose the compound variable we train two intermediate architectures, each changing exactly one ingredient of CNN-4 while holding the dataset (CIFAR-10), training recipe, depth, widths and layer names fixed: *CNN-4+BN* adds batch normalisation after every convolution (no skips), and *CNN-4+skip* adds a skip connection around every convolution (no normalisation; channel changes are bridged by 1×1 projections, which are not 3×3 convolutions and therefore stay outside the per-filter screen). Both are trained over the same three seeds and screened identically.

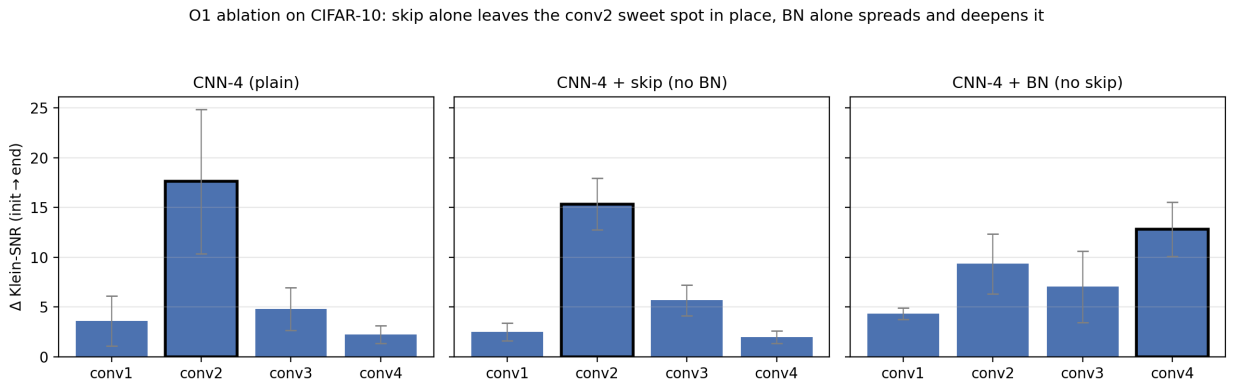

Figure 3: O1 ablation: per-filter ΔSNR per convolution for plain CNN-4, CNN-4+skip (no BN) and CNN-4+BN (no skip), all on CIFAR-10 with the identical recipe (means over three seeds; error bars are std; sweet-spot layer outlined). Skip connections alone leave the `conv2` sweet spot in place; batch normalisation alone spreads the organisation and moves the peak deeper, to `conv4`. Neither alone reproduces the ResNet's relocation to the first post-stem layer (Figure 2), which is therefore emergent from their combination.

The two ingredients act in sharply different ways (Figure 3). *Skip connections alone are inert*: the CNN-4+skip profile is nearly identical to the plain CNN-4, with the sweet spot at `conv2` in all three seeds (ΔSNR $= +15.4 \pm 2.6$, against $+2.5 \pm 0.9$, $+5.7 \pm 1.5$ and $+2.0 \pm 0.6$ at the other layers). *Batch normalisation alone is the active ingredient, but it acts in the opposite direction to the ResNet*: it raises the topological growth of every layer (all four between $+4.3$ and $+12.8$), de-localises the profile, and moves the peak *deeper*, to `conv4` ($+12.8 \pm 2.7$; per-seed sweet spot at `conv4` in two seeds and `conv3` in the third), not earlier. Neither variant reproduces the ResNet's relocation to the first post-stem layer.

The early sweet spot of the residual network is therefore an *emergent* effect of the combination: batch normalisation supplies the organising pressure, and only together with the skip paths (and the depth they enable) does that pressure settle on the earliest 3×3 layer. The scope of this decomposition is the ingredient pair at fixed depth: CNN-4+BN and CNN-4+skip have four convolutions like CNN-4, whereas ResNet-18 has seventeen, so the interaction with depth remains entangled in the final step from CNN-4+BN to the full residual architecture.

We read this relocation, tentatively, as *dynamic capacity allocation*: the architecture concentrates topological organisation at the *earliest* layer able to support it, with batch normalisation making an early layer able to. This is consistent with progressive feature specialisation (O3) and, loosely, with information-bottleneck accounts of depth (Shwartz-Ziv & Tishby, 2017; Santurkar et al., 2018b). We stress it is an *interpretation*, not a measured mechanism: our topological descriptors are an observational proxy for compression, not a computation of mutual information, which we leave to future work.

The sweet spot is a high-learning-rate phenomenon. It is moreover a property of the feature-learning (high-learning-rate) regime, not of depth or of the degree of fitting. Under a low-learning-rate recipe CNN-4 loses the sweet spot while fitting *more*, and a depth-10 2×2 factorial (CNN-10, plain /+BN/+skip/+BN+skip) recovers it only at the high learning rate that batch normalisation makes accessible, in the spirit of the lazy-versus-rich distinction (Chizat et al., 2019); even then the organisation settles *late* rather than at the first post-stem layer, so ResNet's stem and stage structure also shape *where* it sits. Part of batch normalisation's role in O1 is thus *enabling*: it makes the regime in which the weights crystallise accessible at depth. Full factorial and regime profiles are in App. C.

5.5 Scaling to a deeper residual network

To test whether these signatures survive greater depth, we train a CIFAR ResNet-34 (33 3×3 convolutions against ResNet-18's 17, widths held fixed so depth is the only change) over five seeds. Both reproduce. The

aggregate sweet spot is again the first 3×3 layer after the stem (`layer1.0.conv1`, ranked first of all 33), the same early position as ResNet-18, though the per-seed localisation is noisier at depth; and the TwoNN intrinsic dimension contracts in all 33 layers and five seeds (strongest drop -203.5 ± 6.8, matching ResNet-18's -228 ± 8.0). The deeper network thus preserves both the architecture-driven *position* of the sweet spot and the dimensional contraction, degrading gracefully rather than disappearing.

6 Observation 2: the LSTM functional sweet spot at the output gate

We apply the same screen to each of the four LSTM gates (i input, f forget, g candidate, o output) of every layer, separately for the input-to-hidden (W_{ih}) and hidden-to-hidden (W_{hh}) weights, 16 gate point clouds per checkpoint for a two-layer cell.

6.1 Experimental setup

A two-layer LSTM (embedding dim 128, hidden dim 128) trained on IMDB binary sentiment classification, three seeds $\{7, 42, 123\}$. We report ΔSNR (per-gate H_1 signal-to-noise) and the change in TwoNN intrinsic dimension, ΔID, between init and convergence (mean $\pm$ std over seeds).

6.2 Result

Among the 16 gates, the *output gate of the deepest layer's input-to-hidden weight*, `l1.ih.o`, has the largest topological growth: $\Delta\text{SNR} = +1.75 \pm 0.83$ (Figure 4) and the strongest and most seed-stable intrinsic-dimension compression, $\Delta\text{ID} = -20.2 \pm 0.88$ (Table 4). Two contrasts make this a *functional* sweet spot rather than a positional one. The output gate's input-to-hidden weight wins only at the deep layer: the same gate at layer 0 barely moves ($\Delta\text{SNR} = +0.11 \pm 0.98$), so it is the gate's *role* combined with depth, not depth alone, that matters. And the effect is signed: the forget gate of the shallow recurrent connection (`l0.hh.f`) actually *loses* H_1 structure (-0.34 ± 0.83), and the candidate gate of the deep recurrent connection (`l1.hh.g`) is the only other gate with a negative ΔSNR (-0.90 ± 1.00), so the screen discriminates between gate roles rather than merely tracking overall weight movement. The runner-up in Figure 4 is the deep recurrent forget gate `l1.hh.f` ($\Delta\text{SNR} = +1.36 \pm 0.86$): its topological growth is comparable to the sweet spot's, but unlike `l1.ih.o` it is not corroborated by the dimension descriptor, with a weaker and seed-unstable compression ($\Delta\text{ID} = -10.3 \pm 5.7$ versus -20.2 ± 0.88), which is why we identify the sweet spot with the deep output gate rather than with it. The intrinsic-dimension drop, by contrast, is universal across all sixteen gates (every ΔID is negative, ranging from -4 to -20), but the deep output gate is again its sharpest and least variable instance.

6.3 Interpretation

The output gate decides "how much of the cell state to expose" to the next layer and ultimately to the classifier, so its deep input-to-hidden weight is the natural locus at which the geometric primitives feeding the decision must organise. The forget gate, by contrast, learns broad memory-retention behaviour that is by design less selective, consistent with its loss of topological structure. Unlike the CNN case, where the sweet spot is a single *layer*, the LSTM concentrates organisation in a specific *gate role*: the right level of analysis for recurrent cells is (layer × kind × gate), not (layer × kind).

7 Observation 3: ViT-mini depth-monotonic FFN compression

7.1 Experimental setup

ViT-mini, a scaled-down Vision Transformer (Dosovitskiy et al., 2021) built from standard multi-head self-attention (Vaswani et al., 2017): 6 transformer blocks, 4 heads, embedding dim 128, feed-forward dim 256, 4×4 patches, on CIFAR-10, three seeds $\{7, 42, 123\}$. Per block we screen three point-cloud mappings: *attn-per-head-vec* (one point per head), *attn-per-head-rows* (rows of each head's W_Q, W_K, W_V), and *ffn-rows*

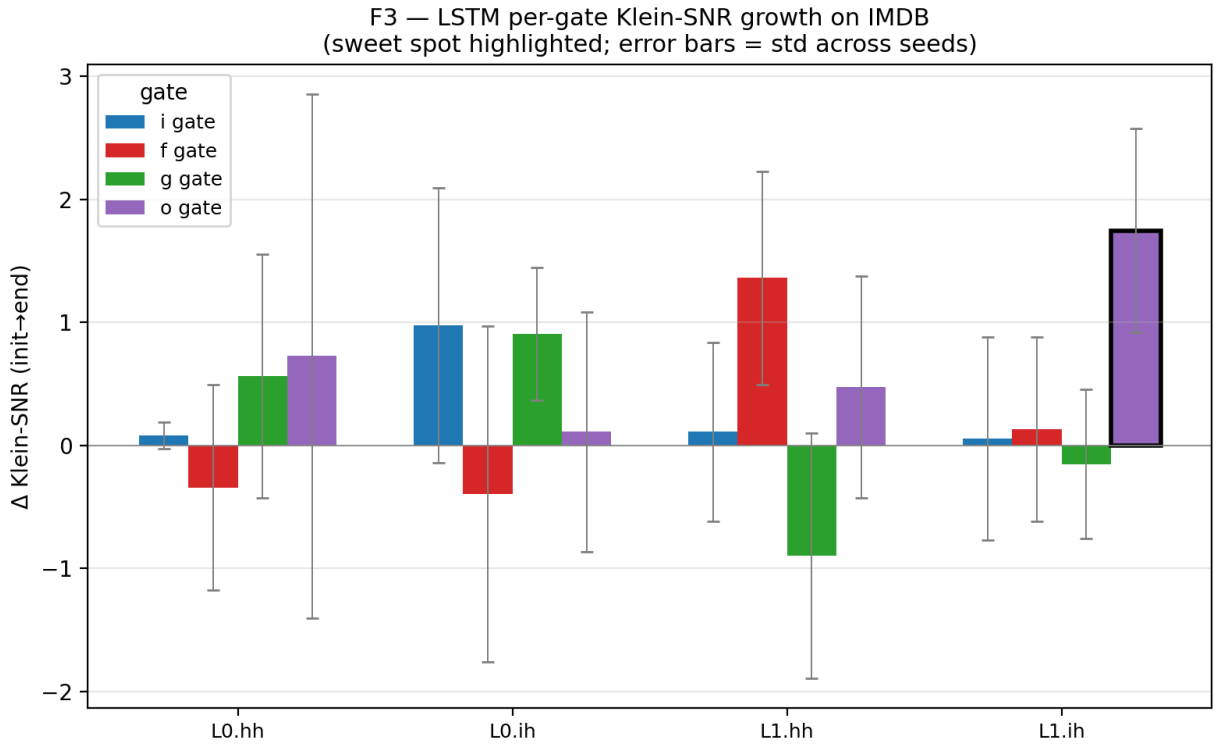

Figure 4: O2: per-gate H_1 signal-to-noise growth on the LSTM, grouped by (layer, weight kind). Bars are means over three seeds; error bars are std; the global sweet spot (deep output gate, `l1.ih.o`) is outlined in black.

(rows of the feed-forward W_1, directly comparable to an MLP layer). We report TwoNN intrinsic-dimension change ΔID and per-head ΔSNR (mean $\pm$ std over seeds).

7.2 Result: deeper feed-forward blocks compress more

The TwoNN intrinsic dimension of the feed-forward weight rows contracts in every block, and the contraction is *stronger in the deeper blocks* (Figure 5a): block 0 $\Delta\text{ID} = -17.8 \pm 3.5$, rising to block 4 -26.5 ± 6.2 and block 5 -21.0 ± 4.8, versus the shallow blocks 1–2 (-15.2 ± 4.5, -16.0 ± 0.9). The trend is monotone in the block average but not strictly per-block (block 5 dips below block 4), so we state it as "deeper blocks (3–5) compress more than shallow blocks (0–2)" rather than strict monotonicity. This contrasts with the activation-ID "hunchback" of Ansuini et al. (2019), where representation ID *rises* then falls with depth: the weight-space ID we measure decreases throughout and accelerates with depth.

The terminal block trades dimensional compression for topological concentration. The dip at block 5 is not noise but a consistent signature of the output-adjacent block, and a second descriptor resolves it (Figure 5). While dimensional compression peaks at block 4, the *persistence entropy* (Eq. 1) of the block-5 FFN cloud collapses far more than any other block ($\Delta\text{PE}_1 = -1.47 \pm 0.43$ versus -0.34 to -0.68 elsewhere), ending at the lowest entropy of the network (4.80 ± 0.45 against ~ 5.9 for the middle blocks), while its normalised top-1 H_1 lifetime rises to 0.028 ± 0.014, three to four times the value of blocks 1–3. The block whose weights feed directly into the classification head thus does *not* continue to reduce its intrinsic dimension the most; it instead concentrates its H_1 structure onto a single dominant cycle. The two ID-like descriptors, spatial TwoNN and topological persistence entropy, decouple at the output, which both explains the terminal anomaly and suggests that proximity to the decision changes *how* the weights organise: from dimensional contraction in the bulk to cycle concentration at the head.

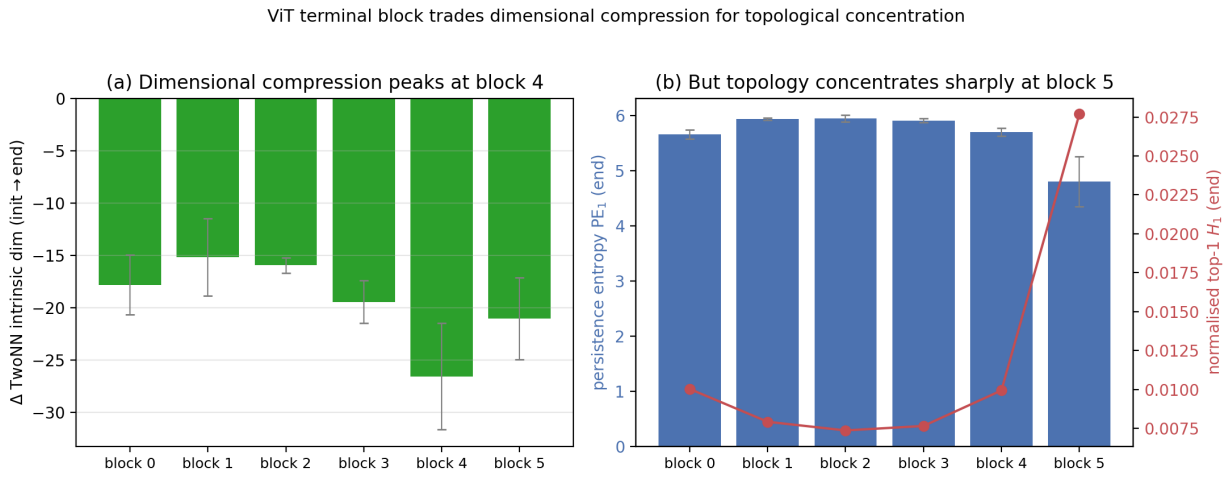

Figure 5: The ViT terminal-block anomaly resolved. **(a)** TwoNN dimensional compression per FFN block peaks at block 4, not block 5. **(b)** Yet the persistence entropy (left axis, bars) of the block-5 cloud collapses and its normalised top-1 H_1 lifetime (right axis, line) spikes: the output-adjacent block concentrates its topology onto a single dominant cycle rather than further reducing dimension. Means over three seeds; error bars are std.

The dominant cycle is isolated by pruning its peers, not grown. The phrase "concentrates onto a single dominant cycle" could describe two very different events: the terminal block could *grow* a new, larger loop, or it could *prune* the surrounding population of cycles and leave a pre-existing leader standing alone. Decomposing the H_1 diagram block by block (Figure 6) shows unambiguously that it is the latter. The lifetime of the single longest cycle is *preserved* at the output block (top-1 $= 0.076 \pm 0.010$), squarely within the 0.071–0.089 range spanned by the other blocks and in fact slightly below block 4's 0.089; no new, more prominent loop is created. What collapses is everything around it: the total number of H_1 cycles falls from a mid-network plateau of $\sim$ 510–527 to 128 ± 11 at block 5, the number of cycles comparable to the leader (lifetime > 0.5 top-1) falls from $\sim$ 65–74 to 18 ± 18, and the total H_1 persistence (the diagram's summed "mass") drops roughly five-fold, from $\sim$ 10.5–11.4 to 2.23 ± 0.62. Because the leader survives while its field is decimated, its *share* of the total persistence ($\ell_{(1)}/L_1$, with L_1 the total H_1 persistence of Eq. 1) quadruples to 0.036 ± 0.015, and the persistence entropy (Eq. 1), a Shannon entropy over the normalised lifetime distribution, collapses to 4.49 ± 0.08 as that distribution concentrates on one bar.

This also explains the decoupling from TwoNN mechanistically. TwoNN estimates dimension from the ratio of first- to second-nearest-neighbour *distances*, a purely local quantity; the pruning we observe removes *global*, long-range loop structure (precisely what H_1 measures) while leaving each point's local neighbourhood essentially intact. The two descriptors decouple at the output not by coincidence but because they probe different scales: local metric geometry, which has little left to compress at the terminal block, and global topological organisation, which undergoes a sharp, late sparsification as the block commits to the decision. The terminal anomaly is thus not a stalling of learning but a *change of regime*, from dimensional contraction in the bulk to topological pruning at the head.

7.3 Distributed per-head Klein-SNR sweet spots

Unlike the CNN case, attention heads show no single dominant layer: per-head ΔSNR has several comparable local maxima spread across blocks, the largest being block 5 head 3 ($+1.31 \pm 0.81$). Head-level topological growth is therefore weak and *distributed*, consistent with multi-head attention's parallel, redundant structure, in which no single head is forced to carry the geometric primitives.

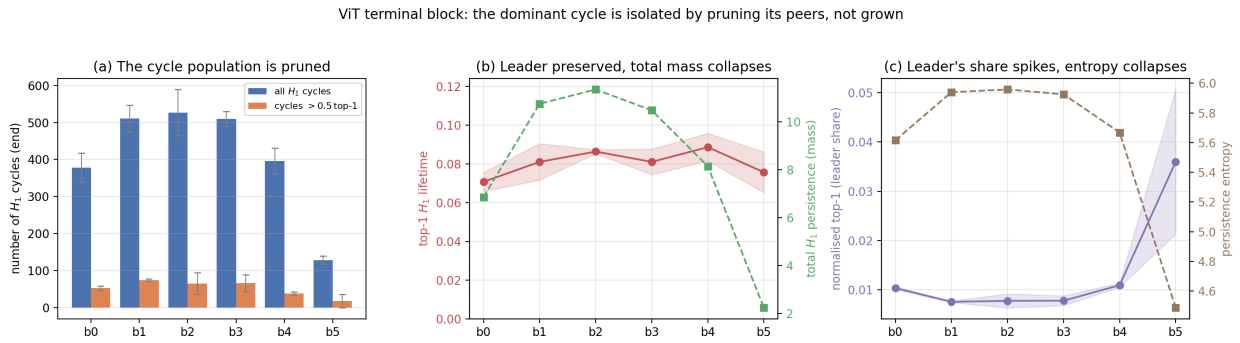

Figure 6: The exact nature of the ViT terminal-block dominant cycle (three seeds; bands/bars are std). **(a)** The cycle population is pruned: both the total number of H_1 cycles and the number comparable to the leader collapse at block 5. **(b)** Yet the single longest cycle's lifetime (left axis) is preserved across blocks while the total H_1 persistence (right axis) collapses. **(c)** Consequently the leader's share of the total persistence (left axis) spikes and the persistence entropy (right axis) collapses. The dominant cycle is isolated by removing its competitors, not by growing.

7.4 Interpretation

The depth-accelerating FFN compression reads as progressive feature specialisation: deeper feed-forward blocks collapse onto lower-dimensional weight manifolds as they commit to higher-level concepts. The distributed per-head signal reflects the opposite pressure in the attention sub-layer, where parallel heads share the representational load. Set beside Ansuini et al.'s activation hunchback, the two are consistent: representation dimension peaks at intermediate depth while weight dimension contracts monotonically, two sides of the same specialisation process, one measured on activations and one on weights.

8 Observation 4: intrinsic-dimension compression across families

A unifying observation across all five families is that the intrinsic dimension of the weight point clouds contracts during training, both spatially (the cloud at each checkpoint) and temporally (the optimisation path that reaches it).

8.1 Experimental setup

For every family we measure the spatial TwoNN intrinsic dimension of the weight clouds at init and at convergence (ΔID, three seeds), reported at each family's strongest-drop layer (Table 4) alongside an MLP zero-init degenerate control. Separately, we re-implemented the Birdal et al. (2021) persistent-homology dimension (Eq. 6) in `tda/phd.py` and computed it on the whole-model weight *trajectory* of all seven families, three seeds each (Table 5).

8.2 Result

Spatially, the TwoNN dimension contracts wherever there is a non-trivial cloud to contract: the strongest layer-level drops span two orders of magnitude in absolute terms but are uniformly negative (Table 4). *Temporally*, the PHD is small in every case (1.0–2.7) and *decoupled from the parameter count*: ResNet-18 with 2.8M parameters has the *lowest* PHD (1.04 ± 0.00), while the 55K-parameter LeNet sits near the top (2.69 ± 0.05). Plain CNNs and the ViT trace $\sim$2.5–2.7-dimensional paths, the LSTM $\sim$1.8, and the residual/BN network and the degenerate zero-init MLP the most contracted paths ($\sim$1.0–1.2).

Family	Strongest-drop layer (mapping)	ΔID (mean ± std)
MLP / MNIST (tiny init)	`fc1` (dense neurons)	-125 ± 21
LeNet / MNIST	conv3 (per-channel)	-47.7 ± 8.6
CNN-4 / CIFAR-10	conv4 (per-channel)	-50.6 ± 92
ResNet-18 / CIFAR-10	layer4.1.conv1 (per-channel)	-228 ± 8.0
LSTM / IMDB	`l1.ih.o` (gate)	-20.2 ± 0.9
ViT-mini / CIFAR-10	block-3 attn (per-head-vec)	-171 ± 123
MLP / MNIST (zero init)	`fc2` (degenerate control)	$+1.3 \pm 0.2$

Table 4: Spatial TwoNN intrinsic-dimension change of the weight clouds (init→end, 3 seeds), at each family's strongest-drop layer. Drops are universal and negative for every genuinely trained model; the zero-init control shows no contraction. Absolute magnitudes are *not* comparable across families (they scale with the mapping's ambient dimension); the table supports the universality of *sign*, not of magnitude.

Family	PHD (mean ± std)	R^2	# params
ResNet-18 / CIFAR-10	1.04 ± 0.00	0.59	2,802,420
MLP / MNIST (zero init)	1.19 ± 0.02	0.50	79,400
MLP / MNIST (tiny init)	1.44 ± 0.01	0.71	79,400
LSTM / IMDB	1.79 ± 0.07	0.98	1,544,448
ViT-mini / CIFAR-10	2.52 ± 0.01	1.00	805,632
LeNet / MNIST	2.69 ± 0.05	0.97	54,544
CNN-4 / CIFAR-10	2.73 ± 0.00	0.99	260,960

Table 5: Birdal persistent-homology dimension of the whole-model weight trajectory (3 seeds each). PHD is small (1–3) and decoupled from the parameter count, and is highly stable across seeds. R^2 is the log–log fit quality of Eq. 6; the low-R^2 rows (ResNet, zero-init MLP) are the most contracted, near-degenerate paths : for such paths the total H_0 persistence $E_0^1(A_n)$ is nearly independent of the subsample size n, so the log–log regression has almost no slope to estimate and its R^2 is necessarily low. The poor fit is itself a symptom of the contraction, not a failure of the estimator.

8.3 Robustness

The MLP zero-init control behaves exactly as it should: with weights pinned at the origin the cloud is degenerate and its dimension does not contract ($\Delta\text{ID} \approx +1.3 \pm 0.2$), confirming that the spatial effect is driven by genuine learning and not by the estimator. The temporal PHD is likewise stable: the cross-seed std is tiny (largest 0.07), so PHD is a property of each architecture rather than of a particular run.

8.4 Interpretation and scope

Together, the spatial TwoNN drop and the temporal PHD give a two-axis picture on the same experiments: the weights occupy a low-dimensional cloud *and* reach it along a low-dimensional path, the explicit contrast our framework was designed to make, and for PHD exactly the message of Birdal et al., here reproduced across five architecture families. The optimisation path therefore explores a low-dimensional fractal set whose dimension is set by the learning dynamics, not by over-parameterisation. The absolute TwoNN magnitudes, however, are *not* comparable across families, since they depend on the ambient dimension of each mapping: Table 4 supports only the qualitative, sign-and-presence claim that spatial compression occurs in every family we test, and the comparable, scale-free version of this statement is what Sec. 7.2 reports per ViT block.

9 Observation 5: discrete-curvature signatures of the weight manifold

The Forman–Ricci curvature (Eq. 3) of the Mapper graph provides a second, geometry-of-the-graph view that is complementary to the persistence of the raw cloud. Two regimes emerge across the three seeds.

9.1 Measured descriptor

The quantity tracked is the init→end change of the mean edge curvature of a layer's Mapper graph, $\Delta\mathrm{Ric}_F$. Positive curvature arises on densely interconnected, hub-like neighbourhoods (sphere-like geometry), negative curvature on branching, tree-like ones (hyperbolic geometry). A *positive shift* during training therefore reads as the layer's weight cloud condensing onto a compact, well-connected manifold, the same event that SNR_1 detects through the emergence of a dominant cycle, but measured on an entirely different construction.

9.2 Result

The CNN sweet-spot layer curves positively. In CNN-4 the first convolution's Mapper graph becomes markedly more positively curved during training, Ric_F moving from $+0.26$ at init to $+1.33 \pm 0.56$ at convergence ($\Delta = +1.07 \pm 0.56$), whereas the deeper conv3 moves the other way, $\Delta\mathrm{Ric}_F = -0.49 \pm 0.56$, becoming hyperbolic. A positive Forman shift signals the appearance of densely connected, hub-like nodes in the Mapper graph, the discrete-geometry counterpart of a cloud condensing onto a low-dimensional, well-connected manifold, while the negative shift in deeper layers signals tree-like branching. ResNet-18 shows the same positive shift concentrated at the first residual 3×3 layer ($\Delta\mathrm{Ric}_F = +1.07 \pm 1.18$), echoing the O1 sweet-spot position.

9.3 Robustness

Degenerate controls behave as expected. The zero-init MLP control collapses to a star-shaped Mapper graph and its Forman curvature drops by the maximal amount the topology allows ($\Delta\mathrm{Ric}_F = -15.0 \pm 0.0$ on `fc2`), confirming the descriptor reacts to genuine geometric change and not to noise; gates and FFN blocks with too few points to form a non-trivial Mapper graph correctly return $\mathrm{Ric}_F \equiv 0$.

9.4 Interpretation and scope

The content of O5 is that the two detectors agree on *where* the condensation happens: the layer with the largest $\Delta\mathrm{SNR}_1$ (O1) is also the layer with the most positive $\Delta\mathrm{Ric}_F$, so the sweet spot is not an artefact of the persistence pipeline. We report Forman curvature throughout because it is $O(|E|)$, whereas the optimal-transport-based Ollivier curvature (Eq. 4) is prohibitively slow on the weight-cloud Mapper graphs (its per-edge Wasserstein solve made a full sweep impractical) and is implemented for targeted small-graph checks only. Curvature thus corroborates O1 from an independent descriptor family: the layer that develops the dominant H_1 cycle is also the layer whose Mapper graph becomes most positively curved.

10 Observation 6: weight-space topology does not predict the generalisation gap (an apparent link is an artefact of unequal fitting)

A natural question is whether the descriptors above, read off the weights alone, also *predict* how well a network generalises. We find that they do not, once the training regime is properly controlled. We report it in detail because it falsifies a concrete, published-style claim, a strong link ($\rho = 0.92$) that collapses to $\rho = 0.10$ once fitting is equalised, and because that falsification is what motivates the object change of O7, where the gap signal does live. The argument has three steps, which the two subsections below and Figure 7 follow in order: (i) a naive label-noise sweep shows a strong apparent correlation between the sweet-spot topology and the generalisation gap (Spearman $\rho = +0.92$); (ii) that correlation is carried entirely by a lurking variable, the train accuracy, which in the naive sweep falls with the noise level and drives *both* quantities (each correlates with it at $\rho = +0.96$); (iii) re-running the sweep in a controlled regime in which every run reaches train accuracy 1.0, so that the gap varies only through genuine generalisation, makes the topological signal vanish ($\rho = +0.10$, n.s.).

10.1 An apparent, but confounded, correlation

We first ran the standard label-noise intervention of Zhang et al. (Zhang et al., 2017): CNN-4 / CIFAR-10 with a fraction $p \in \{0, 0.25, 0.5, 0.75, 1.0\}$ of training labels randomised (two seeds, ten runs), recording the generalisation gap $g = \mathrm{acc}_{\mathrm{train}} - \mathrm{acc}_{\mathrm{test}}$ and, separately, the converged `conv2` descriptors. In this sweep the topological signal-to-noise of the dominant `conv2` H_1 cycle tracks the gap closely: Spearman $\rho = +0.92$ ($p < 10^{-3}$; 95% bootstrap CI $[+0.55, +1.00]$; Fig. 7a), with persistence entropy ($\rho = -0.84$) and normalised top-1 lifetime ($\rho = +0.92$) giving the same picture. On its own this would pass for a weight-only predictor of generalisation.

The result is an artefact of the *training regime*. That sweep ran for only 20 epochs at a high constant learning rate, so the network never fits the noised labels: its train accuracy *falls* with noise (from 0.83 at $p = 0$ to 0.22 at $p = 1$). Train accuracy is therefore a lurking variable that drives both quantities: across the sweep SNR_1 and the gap each correlate with train accuracy at $\rho = +0.96$, and their mutual $\rho = 0.92$ is exactly what that shared dependence predicts. The colour coding in Fig. 7a makes this visible: the low-SNR, low-gap points are precisely the under-fit, high-noise runs.

10.2 The controlled test

To remove the confound the network must reach the *same* (high) train accuracy at every noise level, so that the gap varies through genuine generalisation rather than through how much of the training set was fit. A short calibration showed that CNN-4 cannot memorise fully-noised CIFAR-10 under the original recipe (train accuracy thrashes and ends at chance), but that a smaller training subset (10k) with a lower learning rate (0.01) and cosine decay reaches train accuracy 1.0 at *every* noise level. Under this recipe we ran a 45-run sweep over three studies: label noise $\{0, \ldots, 1\} \times 5$ seeds (subset 10k, no weight decay); a train-size study (subset $\{5, 20\}$k at $p = 1.0$); and a weight-decay study ($\{5 \times 10^{-4}, 5 \times 10^{-3}\}$ at $p = 1.0$). The noise study yields a clean, tight gap spread driven purely by test accuracy: train accuracy is 1.0 throughout while the gap rises monotonically from 0.374 ± 0.008 at $p = 0$ to 0.900 ± 0.004 at $p = 1$ (five seeds).[1]

With the regime controlled, the topological signal disappears. Regressed against this clean gap, the `conv2` Klein SNR is *flat* and uncorrelated (Spearman $\rho = +0.10$, $p = 0.63$, $n = 25$; 95% bootstrap CI $[-0.34, +0.51]$, straddling zero; Fig. 7b), as are normalised top-1 ($\rho = +0.01$), total persistence ($\rho = +0.21$) and TwoNN dimension ($\rho = -0.06$); persistence entropy is at best marginal ($\rho = +0.40$, $p = 0.05$). The two robustness studies confirm what the descriptor actually tracks. Varying the train-set size at fixed gap moves the SNR substantially (from 6.3 at 5k to 10.3 at 20k) while the gap stays at ≈ 0.90, so SNR follows the size and richness of the weight cloud, not generalisation. And strong weight decay (5×10^{-3}), which prevents the network from fitting the random labels (train accuracy back to 0.11), collapses the SNR ($8.6 \to 4.6$) and the gap ($0.90 \to 0.01$) *together*; once more the degree of fitting moves both.

10.3 Non-topological baselines

The collapse is not specific to topology. On the same controlled sweep we subjected two families of non-topological descriptor to the identical protocol: loss-landscape *sharpness* (the mean training-loss increase under random parameter perturbations of radius $\rho = 0.05$), and linear-algebraic summaries of the *same* `conv2` filter cloud the screen reads, namely its spectral and Frobenius norms, stable rank, effective rank and participation ratio, the second-order (covariance-spectrum) counterparts to the persistence and intrinsic-dimension descriptors.

In-distribution, some baselines correlate more strongly than the topology. Pooled across the 25 runs, sharpness tracks the gap (Spearman $+0.72$, $p < 10^{-4}$) and so do several cloud descriptors, more strongly than the flat topological signal-to-noise ($\rho = +0.10$): the stable rank reaches $\rho = +0.91$ and the

[1] Defining the gap *relatively*, $g/\mathrm{acc}_{\mathrm{train}}$, changes nothing in this controlled regime: train accuracy is 1.0 at every noise level by design, so the absolute and relative gap coincide. In the naive sweep of the previous subsection a relative gap would rescale the points, but both its numerator and denominator are driven by the degree of fitting, so the confound remains.

participation ratio $\rho = +0.79$. Taken on its own this would read as a non-topological weight predictor of the gap.

It is the same fitting confound, shown two ways. First, under the honest leave-one-noise-level-out *extrapolation* that O6 demands, the apparent signal mostly vanishes: sharpness ($R^2 = -0.44$), the Frobenius norm (-0.76), the stable rank (-0.16) and the effective rank (-0.12) all fail to predict the gap at a held-out noise level, as the topological descriptors do; only the spectral norm ($R^2 = +0.12$) and the participation ratio ($+0.35$) keep a weak partial signal. Second, and decisively, the train-size control isolates what that residual tracks: holding the gap fixed at ≈ 0.89 while varying the training-set size from 5k to 20k (at $p = 1.0$), the descriptors that had correlated with the gap no longer do, the participation ratio even reversing sign ($\rho = -0.49$ with the gap) and the stable rank going flat ($\rho = +0.07$), while both, like the topological SNR (which moves $6.3 \to 10.3$), instead track the *size and richness* of the weight cloud. The in-distribution correlation was thus collinear with the gap only *within* the noise sweep, where the gap and the amount of randomness the network absorbs rise together; it is not a label-free read of generalisation.

Conclusion. None of the weight-*cloud* descriptors we test, topological or linear-algebraic, predicts the generalisation gap once fitting is equalised and the cloud-richness confound is controlled: the persistent-homology signal-to-noise of the weight cloud, like the linear baselines, tracks *how much of the data the network has fit* and *how rich the weight cloud is*, not the gap. The concrete consequence is a specific result, not a methodological reminder: the prior weight-topology literature relates topological structure to generalisation without equalising how much each network has fit, and under exactly that protocol we reproduce a strong link ($\rho = 0.92$); equalising train accuracy collapses it to $\rho = 0.10$ (n.s.). A published-style claim is therefore an artefact of the uncontrolled fitting, and we can say so with the number attached. Stated plainly, the apparent link is a capacity-saturation confound: it conflates interpolation across networks fit to differing degrees with genuine extrapolation to held-out error. We accordingly propose the controlled protocol, equalise train accuracy and then test under extrapolation, as the standard a weight-to-gap claim should meet. We are careful about the scope of this negative: it concerns descriptors of the weight *cloud*'s geometry, not the broader programme of weight-based generalisation theory. Flatness and PAC-Bayes accounts make claims of a different, bound-theoretic character that our screen does not adjudicate; our sharpness comparison is one empirical probe that a popular flatness measure fails the same extrapolation test, not a verdict on loss-landscape geometry at large. This does not weaken O1–O5: those are claims about *how* and *where* weight geometry organises, a question logically separate from prediction, and they never relied on a gap correlation. Where the gap signal *does* live is the subject of O7.

11 Observation 7: held-out representation topology predicts the generalisation gap

The point of what follows is not a new label-free estimator of the gap, which the literature already has, but a controlled *dissociation*: the *same* screen and the *same* networks, moved from the weights to the representations, turn a non-predictor into a predictor, isolating *where* the signal lives. O6 is negative because the weights are, in effect, a record of how far the network has fit the training data, which is what the loss already measures. A predictor of held-out error must instead read an object that the training loss does not determine. The geometry of the representations the network induces on *held-out inputs* is such an object: it is computed without labels, yet it reflects how the network organises data it was not trained on.

11.1 Experimental setup

We reuse the controlled design of O6 (CNN-4 / CIFAR-10, subset 10k, lr 0.01 cosine, train accuracy = 1 at every noise level, so the gap varies only through test accuracy) and add two settings to probe generality: CNN-4 on CIFAR-100 (a different dataset) and a CIFAR-adapted ResNet-18 (a different architecture, with batch normalisation and skip connections), each over a graded noise sweep $p \in \{0, 0.25, 0.5, 0.75, 1\}$ with three seeds. For each converged network we push a fixed batch of 512 held-out *test* images through it, take the activations of each 3×3 convolution, average-pool them to one vector per image, and apply the same point-cloud screen (Sec. 3) to the resulting held-out activation cloud. This follows the activation-topology line of Naitzat et al. (2020), but ties the manifold geometry to the generalisation gap rather than

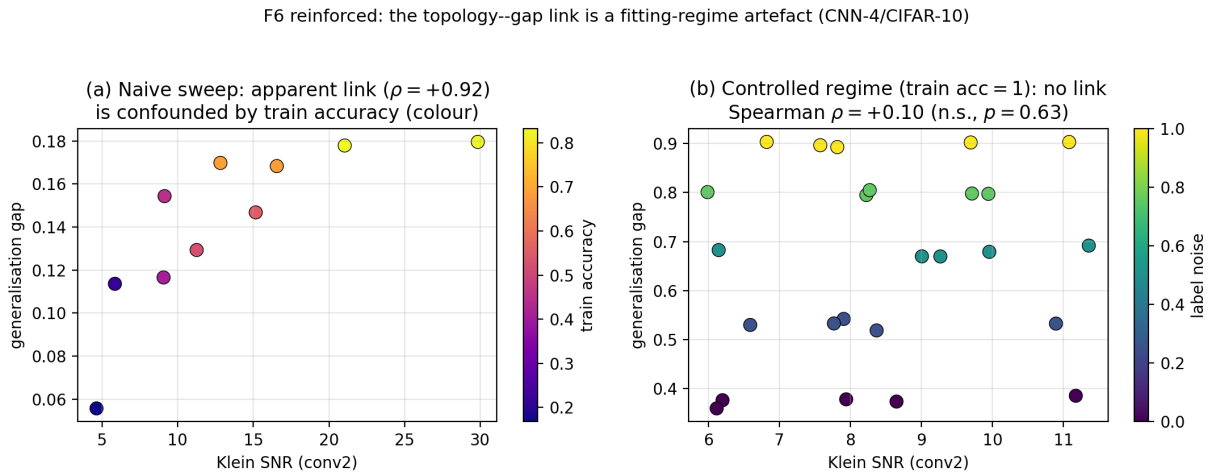

Figure 7: **O6: the weight-topology–gap link is a fitting-regime artefact** (CNN-4 / CIFAR-10). **(a)** The original 20-epoch sweep shows the apparent $\rho = +0.92$, but colouring by train accuracy reveals the confound: SNR and gap both track the degree of fitting (each $\rho = 0.96$ with train accuracy), so their mutual correlation is spurious. **(b)** In the controlled regime (subset 10k, lr 0.01 cosine; train accuracy = 1 at every noise level, 5 seeds, 25 runs) the `conv2` Klein signal-to-noise is flat against the generalisation gap: Spearman $\rho = +0.10$ (not significant, $p = 0.63$); colour is label noise.

to depth. We then ask the question the weight descriptors failed (Sec. 10): does a descriptor predict the gap under *extrapolation*? We regress descriptor on gap with leave-one-noise-level-out cross-validation (fit on four noise levels, predict the held-out one), reporting R^2 and Spearman of the held-out predictions and a label-permutation null for the multivariate model.

11.2 Result

Held-out representation topology predicts the gap, and it extrapolates (Figure 8). A single descriptor already does most of the work: the per-layer intrinsic dimension (TwoNN) of the held-out activation cloud reaches a leave-one-noise-out R^2 of 0.85 (CIFAR-100, `conv1`) and 0.79 (ResNet-18, last residual stage), with H_0 persistence giving 0.71 on the CIFAR-10 reference (`conv3`); a multivariate ridge over all activation descriptors gives $R^2 = 0.82$, 0.84 and 0.79 for the three settings respectively (Spearman 0.87–0.96, permutation $p \leq 0.004$). The signal is consistent: a memorising network leaves its held-out representation manifold higher-dimensional and topologically richer than a generalising one, and the difference grows smoothly with the gap, which is why it extrapolates to noise levels never seen during fitting. This is *extrapolation*, not in-distribution recognition: the predictor is never shown the noise level it must rank, and (as we show below) a version calibrated on CNN-4 transfers *zero-shot* to ResNet-18, ranking a different architecture's gap at Spearman 0.88 without any refitting. The held-out representation geometry is therefore not an artefact of one training set or one convergence trajectory but an architecture-transferable signature of generalisation.

The topology is not redundant with the geometric dimension. Persistent homology carries an independent, additive signal beyond intrinsic dimension. Under the same leave-one-noise-out protocol, adding the topological descriptors raises the extrapolation R^2 over intrinsic dimension alone in every setting (by +0.20, +0.16 and +0.07), and persistent homology alone matches or exceeds it: a memorising network leaves a held-out manifold that is not only higher-dimensional but more topologically *fragmented*, two distinct axes of the same degradation (App. F).

This holds exactly where the weight-space signal of O6 did not. The same controlled regime that flattened the weight descriptors leaves the held-out activation descriptors predictive, and the predictor survives a change of dataset (CIFAR-100) and of architecture to ResNet-18, whose batch normalisation and skip connections

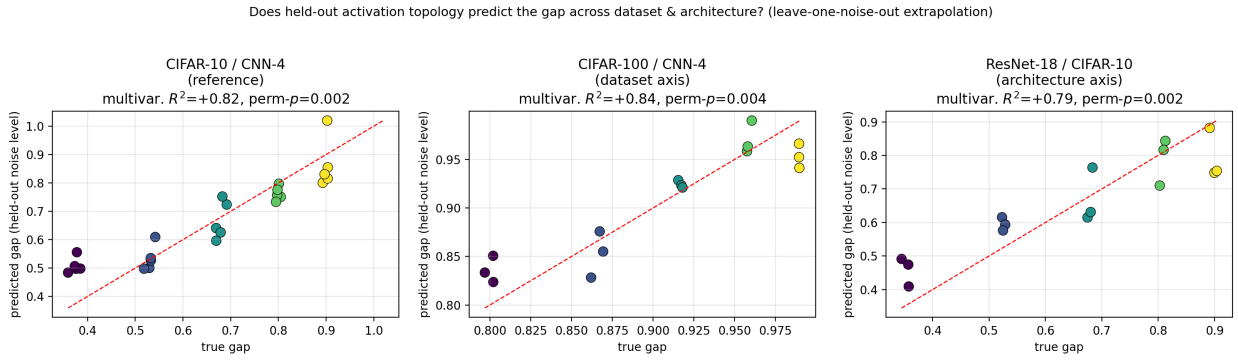

Figure 8: **O7: held-out representation topology predicts the generalisation gap, under extrapolation and across dataset and architecture.** Predicted vs. true gap from a leave-one-noise-level-out ridge over held-out activation-manifold descriptors (colour is the held-out noise level). The predictor extrapolates to unseen noise levels on all three settings: CIFAR-10 / CNN-4 ($R^2 = 0.82$), CIFAR-100 / CNN-4 ($R^2 = 0.84$) and ResNet-18 / CIFAR-10 ($R^2 = 0.79$; permutation $p \leq 0.004$), in contrast to the weight-space descriptors of O6, which do not.

reorganise weight geometry early (O1, O5). The object that carries the generalisation signal is therefore not the weights but the representation they induce on unseen data.

The signal is in the representation map, not in the probe. The predictor does not depend on the choice of held-out inputs. Re-running the screen on 512 *training* images instead of held-out ones, everything else identical, predicts the gap essentially as well (extrapolation R^2 of 0.80 against the held-out 0.82): even where the network reaches accuracy 1 at every noise level, fitting noised labels still distorts the activation geometry, whereas the weights at the same checkpoints carry none of it (O6). The gap signal is thus a property of the learned *representation map*, and the dissociation we report is between objects (parameters vs. representations), not between which inputs are shown; a practical corollary is that the predictor needs only unlabelled inputs the model already has.

A mechanism-invariant component, localised to the early layer. The strongest test of whether the predictor captures generalisation, rather than a label-noise artefact, is to calibrate it on one gap-inducing mechanism and apply it, with no refitting, to another. Train-set size provides a clean second mechanism (at subsets of 2–20k the network still reaches train accuracy 1.0, so the gap varies only through test accuracy): the `conv1` held-out intrinsic-dimension predictor, fit on the label-noise runs, ranks the train-size gaps it never saw almost perfectly (Spearman 0.97, $p < 10^{-3}$; Figure 9a). Along a second, independent axis it transfers *zero-shot* across architectures: calibrated on CNN-4 and applied with no refitting, it ranks ResNet-18's label-noise gap at Spearman 0.88 ($p < 10^{-3}$), exceeding ResNet-18's own self-calibration. In both transfers it is the *ranking* that is invariant, not the absolute scale, and the signal is *localised* to the early representation: the deeper, more task-specific descriptors flip sign across mechanisms (Figure 9b). We test each axis singly and the targets are small (12–15 runs), so the robust content is the ranking invariance and its early-layer localisation, not the precise magnitudes.

Validation on real distribution shift (CIFAR-10-C). The strongest practical test is whether the held-out topology tracks degradation under *real*, documented shift. On CIFAR-10-C (Hendrycks & Dietterich, 2019) (15 corruption types $\times$ 5 severities, a third gap-inducing mechanism never used to calibrate the predictor), a label-free read of the held-out manifold ranks the accuracy drop under leave-one-corruption-out extrapolation: the H_1 total persistence reaches Spearman 0.61 on CNN-4 ($p < 10^{-3}$) and 0.58 on ResNet-18, and a predictor formed on three corruptions confirms on the twelve held out ($R^2 = 0.66$). Two qualifications: the signal is *topological, not geometric* (the intrinsic-dimension predictor of the label-noise setting is flat or sign-inverted here, $\rho = -0.25$, the same ID-vs-PH dissociation, Fig. 20, sharpened under real shift); and it transfers as a *within-model ranking*, the locus being architecture-specific. Full per-layer results, including

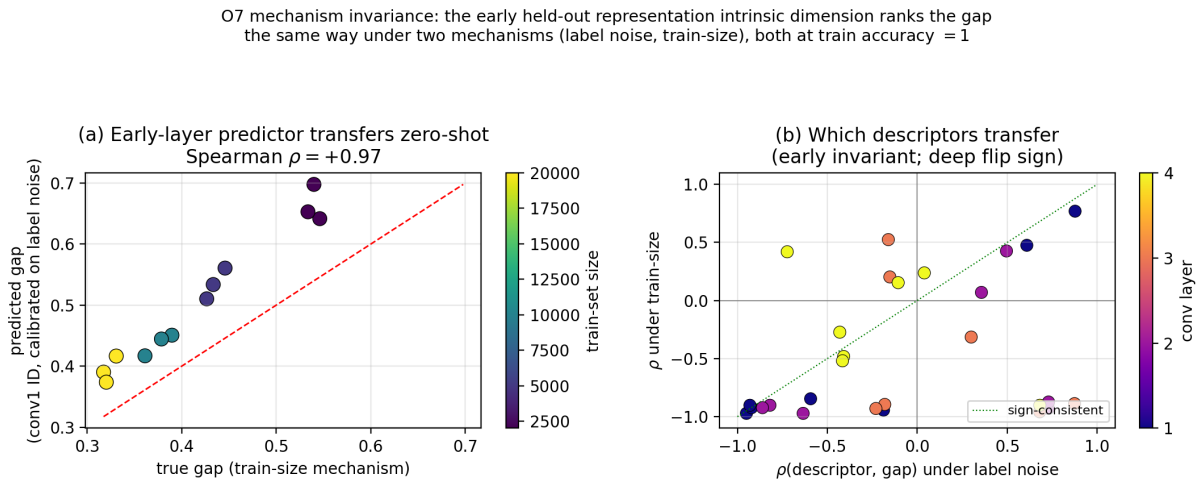

Figure 9: O7 mechanism invariance. **(a)** The `conv1` held-out intrinsic-dimension predictor, calibrated on the label-noise gaps, ranks the train-size gaps it never saw (Spearman 0.97); the red line is identity, and the systematic offset is the non-transferring calibration. **(b)** Per-descriptor gap correlation under label noise (horizontal) versus train-size (vertical), coloured by convolution depth: early descriptors lie on the sign-consistent diagonal, deeper ones flip sign, so the mechanism-invariant signal is localised to the early representation.

the H_2 check, are in App. G. This places the held-out screen in the family of label-free accuracy estimation under shift (Deng & Zheng, 2021a; Guillory et al., 2021; Garg et al., 2022).

Scope. This remains a controlled label-noise study (three settings, 3–5 seeds, $n = 15$–25); the predictor reads unlabelled held-out inputs, the standard setting for estimating generalisation without test labels. The multivariate model is guarded by the permutation null, and the single intrinsic-dimension descriptor is the cleaner, interpretable evidence. The predictor is moreover not specific to the label-noise mechanism, which the next paragraph makes precise. We note that regularisation knobs do not provide a clean alternative test: varying weight decay or augmentation lowers the gap mainly by lowering train accuracy, so they re-enter the O6 fitting confound rather than vary the gap at matched fit, whereas train-set size and label noise both hold train accuracy at 1.0.

12 Discussion

Summary. Across five architecture families and three seeds, the shape of the weights evolves during training in structured, architecture-dependent ways that a single unified screen makes visible. In CNNs a per-filter topological *sweet spot* (peak $\Delta\mathrm{SNR}_1$) sits at a layer whose position shifts with normalisation and skip connections (O1), and that same layer is where the Mapper graph curves most positively (O5). LSTMs relocate the phenomenon from a layer to a *gate role*, the deep output gate (O2). Our ViT-mini compresses feed-forward weight dimension more in deeper blocks while spreading attention-head organisation thinly (O3). Underlying all of them, the TwoNN intrinsic dimension of the weight clouds contracts during training (O4), and the trajectory that reaches those clouds is itself low-dimensional and decoupled from the parameter count (Table 5). We also tested, and rejected, the hypothesis that these descriptors *predict* generalisation: a strong apparent correlation between the `conv2` topological signal-to-noise and the generalisation gap turns out to be an artefact of the training regime, vanishing once train accuracy is held fixed across a controlled 45-run sweep (O6). The signal does live somewhere, though: the topology of the representations a network induces on held-out inputs predicts the gap under extrapolation, across datasets and architectures (O7).

12.1 Implications for generalisation

Two of our observations (O4, and O1/O5) connect to the geometric account of why over-parameterised networks generalise; a third marks the limit of the weight-space connection (O6); and a fourth locates where the signal actually lives, in the held-out representations rather than the weights (O7).

The first is the *two-axis low-dimensionality* of learning that our screen exposes on the same runs. Spatially, the weight clouds contract during training in every family (O4); temporally, the optimisation path that produces them has a persistent-homology dimension of only 1–3, and this dimension is essentially *independent of the number of parameters*: the 2.8M-parameter ResNet-18 traces the lowest-dimensional path (PHD = 1.04) while the 55K-parameter LeNet traces one of the highest (2.69; Table 5). This is precisely the regime in which the generalisation bound of Birdal et al. (2021), which scales with the trajectory's fractal dimension rather than the ambient parameter count, is non-vacuous, in line with the broader programme of complexity-based, rather than parameter-count-based, generalisation measures (Neyshabur et al., 2017; Jiang et al., 2020; Dziugaite & Roy, 2017). Our contribution here is to show that the small, parameter-count-decoupled dimension they tie to generalisation is not specific to their setting but recurs across MLP, CNN, ResNet, LSTM and ViT, and that it is mirrored by an independent *spatial* contraction. That the trajectory should be low-dimensional is consistent both with the existence of sparse trainable subnetworks (Frankle & Carbin, 2019) and with the observation that training confined to a low-dimensional random subspace already suffices (Li et al., 2018a); that the loss surface itself is geometrically benign in such regimes (Li et al., 2018b; Keskar et al., 2017) offers a complementary view of why the path stays simple. The effective complexity of a trained network, measured two different ways on the weights, is therefore far below what its parameter count would suggest. As a practical guideline for quantifying that effective complexity, we recommend the trajectory PHD as the first measurement: it requires only the weight checkpoints already saved during training, costs a minimum-spanning-tree fit over subsamples (Eq. 6), is extremely stable across seeds (std ≤ 0.07 in Table 5) and is decoupled from the parameter count; the per-layer TwoNN ΔID is the complementary spatial reading, useful when one wants to localise *which* layers carry the compression rather than score the model as a whole.

The second is that the *sweet spot marks where generalisable structure is laid down.* Brüel Gabrielsson & Carlsson (2018) established that the prominence of the low-dimensional filter manifold correlates with a CNN's ability to generalise; our O1/O5 results localise *where* in the network that manifold consolidates, the layer whose dominant H_1 cycle grows most and whose Mapper graph curves most positively, and show that batch normalisation and skip connections shift its position. Interpreted in terms of implicit regularisation (Ansuini et al., 2019), this suggests these architectural components do not change *whether* the network forms low-dimensional geometric primitives but *where* along its depth it does so: in a plain network the primitives must accrete through depth and only crystallise late, whereas batch normalisation lets them form immediately after the stem, freeing subsequent layers to specialise. On this reading, the position of the sweet spot indicates how an architecture allocates its generalisable capacity.

The third observation marks the limit of this connection. One might hope that the sweet-spot descriptor, beyond *marking* where generalisable structure forms, also *predicts* how well the network generalises, and a naive label-noise sweep appears to deliver exactly that (Spearman $\rho = 0.92$ between the `conv2` topological signal-to-noise and the generalisation gap). As O6 (Sec. 10) shows, that correlation is a confound: it is carried entirely by train accuracy, which the short sweep let fall with noise, and it vanishes ($\rho = 0.10$, n.s.) once a controlled 45-run sweep holds train accuracy fixed at 1.0 across all noise levels. What the descriptor actually tracks is the degree of fitting and the richness of the weight cloud, not held-out error. The substance of this negative is a falsification, not a caution: the apparent $\rho = 0.92$ is, on the same networks, an artefact of unequal fitting, and holding the fitting fixed leaves no weight-space signal behind. The value of the result is that the falsification is not the end of the story but sets up the dissociation of O7, where the *same* screen, moved to the held-out representations, recovers a predictor the weights do not contain. Whether any weight-space descriptor predicts the gap under properly controlled fitting remains open.

The fourth observation answers that question by changing the object. Weight-space descriptors are redundant with the loss because the weights record how much was fit; the geometry a network imposes on *held-out* inputs is not so determined. Screening the held-out activation manifold (O7, Sec. 11) yields a label-free

descriptor whose per-layer intrinsic dimension predicts the generalisation gap under leave-one-noise-level-out extrapolation ($R^2 \approx 0.8$), and the prediction survives a change of dataset (CIFAR-100) and of architecture (ResNet-18), where the weight-space signal did not. A memorising network leaves its held-out manifold higher-dimensional and topologically richer than a generalising one. The signal the weights and the loss do not carry is present in the representations, which points to representation geometry, not weight geometry, as the right object for topological accounts of generalisation.

13 Conclusions, practical implications and scope

Read together, our results say that weight topology is a descriptor of the training *process* (degree of fitting, where and when structure forms), not of held-out error. That scoping suggests concrete uses, distinct from the generalisation prediction that O6 rules out.

Design guidelines. Three statements transfer to practice. First, when probing, pruning or extracting features one should target the architecture-specific sweet-spot layer, and expect its position to move: batch normalisation and skip connections move it to the first post-stem layer, while plain networks form their geometric primitives late (O1, O5). Second, anything touching generalisation should not be read off the sweet-spot layer alone: it is where structure is strongest but, as O6 shows, the layer least informative about the gap, whereas a distributed multi-layer read carries what little signal exists. Third, and most usefully as a caution, weight topology should not be used as a stand-in for held-out error or as an early-stopping criterion, and any proposed weight-to-generalisation metric must be validated with the training regime controlled and tested for extrapolation rather than in-distribution recognition.

Broader impact: label-free monitoring under shift. The O6/O7 dissociation has a deployment reading the design guidelines above do not capture. The quantity a practitioner most wants, namely whether the model will stay accurate on the data it is now seeing, is what the weights and the loss do not carry (O6), yet a label-free read of the held-out representation geometry recovers it (O7), and continues to do so under real distribution shift (the CIFAR-10-C validation of Sec. 11; App. G). Because it needs only unlabelled forward passes, it is aimed at the bottleneck of deployed learning, where inputs are abundant but labels are scarce or delayed, and it suggests concrete uses: a silent-degradation alarm on a live unlabelled stream, triage of low-reliability batches to human review, and model selection under shift, none of which need a held-out labelled set or data egress, hence compatible with on-premise deployment. Two qualifications govern its use: the descriptor is persistent homology, not intrinsic dimension, and it must be calibrated once per deployed model (its locus and scale are architecture-specific), after which monitoring is label-free. We state these as uses, not guarantees: the estimate is a ranking and an early warning, the magnitudes are indicative, and a nominal-severity oracle (unavailable in deployment) can match a single descriptor on a strong network.

Table 6 collects the seven observations with the evidence tier each rests on, separating the controlled claims (O1, O4–O7) from the relative, trend-level ones (O2, O3).

Limitations. The cross-*family* statements (CNN vs LSTM vs ViT) still rest on relative, scale-free descriptors rather than matched conditions, since each family is trained on the dataset natural to it; within the CNN family, however, the four-dataset study (Sec. D) shows the sweet-spot position is driven by architecture rather than task complexity, and the CNN-4-vs-ResNet leg (Sec. 3.5) is fully controlled, with the BN/skip ablation (Sec. 5.4) further decomposing that compound variable into its components at fixed depth. The depth-10 factorial adds a scoping statement: the sweet spot itself exists only in the high-learning-rate (feature-learning) regime, so the position claims of O1 are claims about networks trained in that regime; at low learning rate the weights still compress (O4) and the networks still fit, but no dominant cycle crystallises. Our architectures are also small to medium by current standards; the ResNet-34 scaling check (Sec. 5.5) shows O1's early sweet spot and O4's contraction persist at roughly twice the depth, though with a noisier per-seed localisation, while larger and pretrained models (deeper ResNets, ViT-L, fine-tuned transformers) remain future work. We are therefore explicit that our claims about *learning and generalisation* are established on representative, non-state-of-the-art models; the most informative next validation is a pretrained model under fine-tuning or transfer, a setting the screen applies to unchanged. The Mapper-derived descrip-

	Claim	Evidence (Table 2)	Key result
O1	CNN per-filter H_1 sweet spot; its layer position shifts from the last convolution (LeNet) to the first post-stem 3×3 (ResNet)	(i) controlled + ablation	relocation isolated to the BN+skip combination
O2	LSTM organisation concentrates in a gate *role* (deep output gate)	(ii) relative, scale-free	largest init→end change at the output gate
O3	ViT-mini feed-forward weight dimension compresses with depth	(ii) relative; (iii) vs Ansuini	deep-block compression trend
O4	Weight-cloud intrinsic dimension contracts, spatially (TwoNN) and along the trajectory (PHD)	(i) controlled, same runs	ΔID < 0 in every family; PHD 1–3
O5	Forman–Ricci curvature flags the same CNN sweet-spot layer	(i) controlled, corroborative	curvature peak coincides with O1
O6	Weight topology does *not* predict the gap once fitting is equalised	(i) controlled (label noise, train acc = 1)	ρ: 0.92 $\rightarrow$ 0.10; non-topological baselines fail too once cloud richness is controlled
O7	Held-out representation topology *does* predict the gap	(i) controlled + external	positive extrapolation across datasets and architectures, and on CIFAR-10-C

Table 6: The seven observations and the strongest evidence tier each rests on (tiers as in Table 2). O1 and O4–O7 rest on controlled, within-paper comparisons; O2 and O3 are relative, scale-free trends, with O3 additionally juxtaposed to a prior activation result.

tors (curvature, cycle counts) do depend on the cover, clustering and lens; we sweep the first two over 27 configurations and replace the third with a learned lens (App. B), finding the trees-vs-surfaces dichotomy invariant and the Forman curvature mean the one descriptor sensitive enough that we keep it corroborative rather than primary. Ollivier–Ricci curvature proved too expensive to run at scale on the weight-cloud Mapper graphs and is used for small-graph checks only, so the curvature evidence (O5) rests on Forman–Ricci. The ViT FFN compression is a deep-versus-shallow trend rather than strict per-block monotonicity. Finally, our generalisation result is negative (O6): we show only that one candidate predictor fails once the training regime is controlled, on a single architecture and dataset, not that no weight-space descriptor could ever predict the gap.

Future work. The negative O6 result invites a properly controlled search for a generalisation predictor: holding train accuracy fixed (as we do here) while varying architecture, regularisation strength and train-set size, and testing descriptors beyond per-layer persistence, including the trajectory PHD that theory most directly ties to the gap. Beyond it: scaling the trajectory analysis to the fine-tuning of pretrained BERT/GPT/ViT-L, where the screen applies unchanged; optimising the Mapper lens for a downstream objective rather than for reconstruction (a reconstruction-learned lens already leaves the conclusions unchanged, App. B); sweeping datasets to test the cross-domain stability of the sweet-spot relocation; and connecting the per-gate and per-head loci to the features and circuits studied in mechanistic interpretability (Olah et al., 2020). More broadly, our per-layer screen is complementary to network-level topological complexity measures such as Neural Persistence (Rieck et al., 2019), and combining the two (a global complexity score with a per-layer localisation of where structure forms) is a promising direction. The screen and all experiments are released as an anonymised public repository (released upon acceptance).

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peirson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nieves Atienza, Rocio Gonzalez-Diaz, and Manuel Soriano-Trigueros. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107, 2020.

Rubén Ballester, Carles Casacuberta, and Sergio Escalera. Topological data analysis of neural network layer representations. *arXiv preprint arXiv:2208.06438*, 2022.

Tolga Birdal, Aaron Lou, Leonidas J. Guibas, and Umut Şimşekli. Intrinsic dimension, persistent homology and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Rickard Brüel Gabrielsson and Gunnar Carlsson. Exposition and interpretation of the topology of neural networks. *arXiv preprint arXiv:1810.03234*, 2018.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

Anton Bushuiev, Daniil Cherniavskii, Eduard Tulchinskii, et al. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models. In *Findings of the Association for Computational Linguistics: EACL 2024*, 2024.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Gunnar Carlsson and Rickard Brüel Gabrielsson. A topology layer for machine learning. *Journal of Machine Learning Research*, 24:1–35, 2023. JMLR 21-0073.

Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76(1):1–12, 2008.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

Ciprian A. Corneanu, Meysam Madadi, Sergio Escalera, and Aleix M. Martinez. Computing the testing error without a testing set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Corinna Coupette, Sebastian Dalleiger, and Bastian Rieck. Ollivier-ricci curvature on graphs: A comparative study. In *International Conference on Learning Representations (ICLR)*, 2023.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning (ICML)*, 2018.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Maxime Gabella. Topology of learning in feedforward neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3588–3592, 2021. doi: 10.1109/TNNLS.2020.3015790.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations (ICLR)*, 2022.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations (ICLR)*, 2022.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, et al. Persistent topological features in large language models. *arXiv preprint arXiv:2410.11042*, 2024.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2004.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018a.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b.

Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal*, 63 (4):605–627, 2011.

P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3: 1236, 2013.

John P. Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning (ICML)*, 2021.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3): 810–864, 2009.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences (PNAS)*, 117(40):24652–24663, 2020.

David Pérez-Fernández and Gregory Naitzat. Characterizing and measuring the similarity of neural networks with persistent homology. *arXiv preprint arXiv:2101.07752*, 2021.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations (ICLR)*, 2019.

Areejit Samal, R. P. Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost. Comparative analysis of two discretizations of ricci curvature for complex networks. *Scientific Reports*, 8(1):8650, 2018.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018a.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Gurjeet Singh, Facundo Mémoli, and Gunnar E. Carlsson. Topological methods for the analysis of high-dimensional data sets and 3d object recognition. In *SPBG*, 2007.

R. P. Sreejith, Karthikeyan Mohanraj, Jürgen Jost, Emil Saucan, and Areejit Samal. Forman curvature for complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(6):063206, 2016.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations (ICLR)*, 2022.

Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Fanghui Xue, Biao Yang, and Jiandong Sun. Searching intrinsic dimensions of vision transformers. *arXiv preprint arXiv:2204.07722*, 2022.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

A Implementation details

All models are trained in PyTorch (≥ 2.3) and are device-agnostic (CUDA / Apple MPS / CPU selected automatically); the experiments here ran on an Apple M1 Pro (16 GB unified memory, MPS backend), with the code validated to run unchanged on Windows 11 + NVIDIA RTX 2000 Ada (16 GB, CUDA). The TDA stack is `ripser/giotto-tda` (persistent homology), `kepler-mapper` (Mapper), `GraphRicciCurvature` (Forman/Ollivier, run single-threaded to avoid a NetworKit OpenMP issue on arm64), `dadapy` (TwoNN), and our own PHD implementation (`tda/phd.py`). Each experiment is trained with three seeds $\{7, 42, 123\}$; weight snapshots are saved every few optimisation steps and sub-sampled to ≤ 12 checkpoints (init→end) per run for the screen. Per-architecture training settings (optimiser, learning rate, batch size, epochs, dataset) are specified in the released `configs/` and run metadata; each run records its git commit, device, and full hyper-parameters for reproducibility.

B Robustness to Mapper hyperparameters and lens choice

A natural concern with any Mapper-based analysis is that the resulting graph, and hence the descriptors read off it, may be artefacts of the cover and clustering hyperparameters. Two points address this. First, our *lead* findings do not use the Mapper graph at all: the per-filter and per-gate signal-to-noise (O1, O2), the feed-forward intrinsic-dimension compression and terminal-block persistence entropy (O3), and the spatial TwoNN and trajectory PHD (O4) are computed from Vietoris–Rips persistence or from nearest-neighbour statistics on the raw point cloud, so they are Mapper-independent by construction. Only the Ricci-curvature corroboration (O5) and the trees-vs-surfaces cycle counts (Sec. 4) read from the Mapper graph.

For those Mapper-dependent quantities we ran a sensitivity sweep over the full grid $n_{\text{cubes}} \in \{10, 15, 20\}$, overlap $\in \{0.3, 0.4, 0.5\}$ and DBSCAN $\epsilon \in \{0.3, 0.5, 0.7\}$ (27 configurations) on the converged sweet-spot filter clouds. The trees-vs-surfaces dichotomy is invariant across the entire grid: the L_2-lens graph has at most 0–2 independent cycles in every configuration of every network, while the PCA-3-lens graph never falls below 106 cycles (LeNet ≥ 536, ResNet ≥ 627), so the two-orders-of-magnitude separation is not a hyperparameter artefact (Figure 10). The Forman curvature mean is more configuration-sensitive (it can change sign for some extreme settings), which is why we present curvature (O5) as corroborative rather than primary evidence.

A learned lens removes the projection-choice doubt. The sweep above varies the cover and clustering but keeps the lens itself *fixed and hand-chosen* (L_2 or PCA-3), so a residual concern is that the trees-vs-surfaces dichotomy is an artefact of those particular projections. We formalise and run the principled mitigation. Define a *learned lens* as a parametric encoder $f_\theta : \mathbb{R}^D \to \mathbb{R}^d$ trained jointly with a decoder g_ϕ to minimise the reconstruction loss $\mathbb{E}\,\|x - g_\phi(f_\theta(x))\|^2$ on the standardised weight cloud; the lens is the d-dimensional bottleneck $f_\theta(X)$. Being data-driven and nonlinear, it is a different family from PCA, so any agreement between the two cannot be a shared-projection artefact. Rebuilding the Mapper graph of each converged sweet-spot cloud under learned lenses of latent dimension 1, 2 and 3 (a small autoencoder, averaged over three seeds) reproduces the dichotomy exactly (Figure 11): the learned 1-D lens gives the same near-tree as L_2 (0 independent cycles for all three networks), while the learned 3-D lens recovers the same surface as PCA-3 at the same order of magnitude (all values are independent Mapper cycle counts $\beta_1(G)$ of the converged sweet-spot cloud) (CNN-4 `conv2`: PCA-3 726 vs. learned 521 ± 43; LeNet `conv3`: 1634 vs. 1388 ± 87; ResNet `layer1.0.conv1`: 1689 vs. 1523 ± 80), with the learned 2-D lens intermediate but still firmly a surface (52–138 cycles). The distinction is thus governed by the *resolving dimension* of the lens, not by the choice of projection: a 1-D lens of any family collapses the surface onto a line, any $\geq$ 2-D lens recovers it. The surface is intrinsic to the weight cloud. The remaining open direction is no longer *whether* a learned lens changes the conclusion (it does not), but whether a lens *optimised for a downstream objective* rather than for reconstruction could localise sweet spots even more sharply, which we leave to future work.

C Training-regime dependence: the depth-10 factorial

This appendix gives the factorial and regime profiles summarised in O1 (Sec. 5.4).

A depth-matched factorial. The BN/skip decomposition of O1 still differs from ResNet-18 in depth, so we repeated it as a full 2×2 factorial at depth 10 (CNN-10: plain, +BN, +skip, +BN+skip; same three pooling stages and stage widths as CNN-4; three seeds; parameter counts matched within 2% across cells). Depth itself forces a recipe change: under the depth-4 recipe the plain and skip-only cells are untrainable, since activations vanish at initialisation under PyTorch's default convolution gain and training diverges at lr 0.05, so all four cells use He initialisation, variance-preserving skip sums $(y + \text{shortcut})/\sqrt{2}$, and lr 0.01, the one recipe that trains them all to comparable train accuracy (0.94–1.00). Under this recipe the sweet spot vanishes *in every cell*: no layer of any cell exceeds $\Delta\text{SNR} = +1.5$ on average and the end-of-training SNR stays at the noise floor (≈ 6) everywhere (Figure 12), even though every cell fits well and the intrinsic-dimension contraction of O4 persists (strongest per-cell drops between -20.0 ± 17.1 and -69.7 ± 7.5).

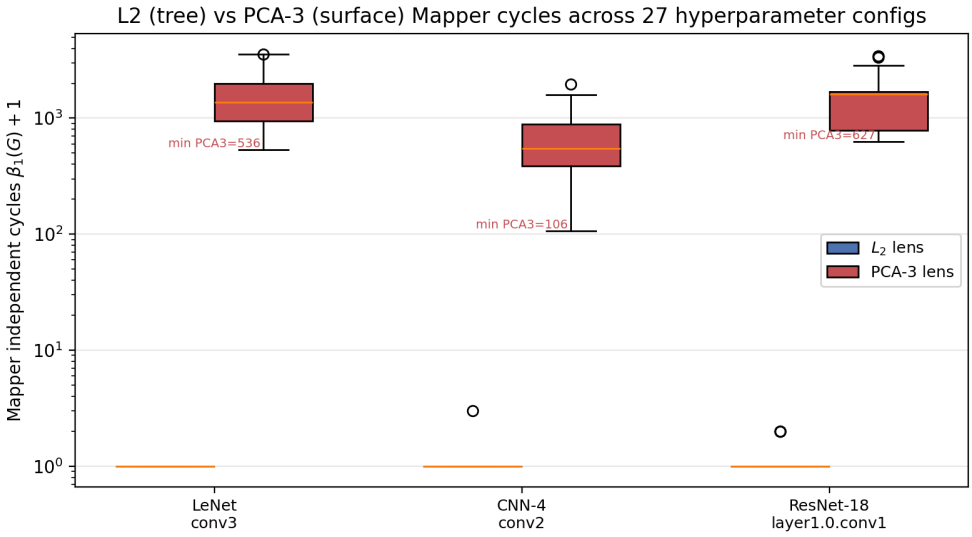

Figure 10: Mapper trees-vs-surfaces dichotomy is robust. Distribution of the independent-cycle count $\beta_1(G)$ under the L_2 and PCA-3 lenses across 27 Mapper hyperparameter configurations, for each CNN sweet-spot layer (log scale, +1 offset). The L_2 box stays at ~ 0 (tree) and the PCA-3 box in the hundreds-to-thousands (surface) for *every* configuration.

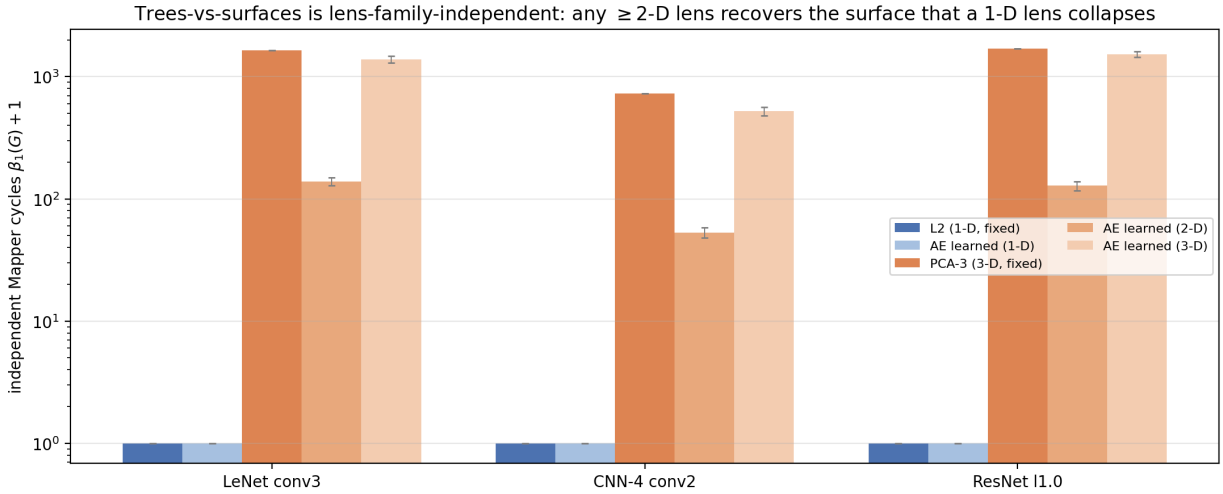

Figure 11: The trees-vs-surfaces dichotomy is lens-family-independent. Independent Mapper cycles $\beta_1(G)$ (log scale, +1 offset) of each converged CNN sweet-spot cloud under five lenses: fixed L_2 (1-D) and PCA-3 (3-D), and *learned* autoencoder lenses of latent dimension 1, 2, 3 (mean $\pm$ std over three autoencoder seeds). Both 1-D lenses (fixed and learned) give a tree ($\beta_1 = 0$); every $\geq$ 2-D lens, PCA *or* learned, recovers the many-cycle surface. The surface is therefore intrinsic to the weights, not a PCA artefact. The fixed lenses (L_2, PCA-3) are deterministic and carry no error bar; the learned-lens bars are means $\pm$ std over the three autoencoder seeds.

The sweet spot is a property of the training regime. A control separates depth from recipe: CNN-4 retrained under the depth-10 recipe (He initialisation, lr 0.01) *also* loses its sweet spot (`conv2` ΔSNR $= +0.86 \pm 1.21$, end SNR 6.7, against $+17.61 \pm 7.27$ and end SNR ≈ 22 at lr 0.05) while fitting *more* (train accuracy 0.96 versus 0.83). The crystallisation of the dominant cycle is therefore not a function of depth, of

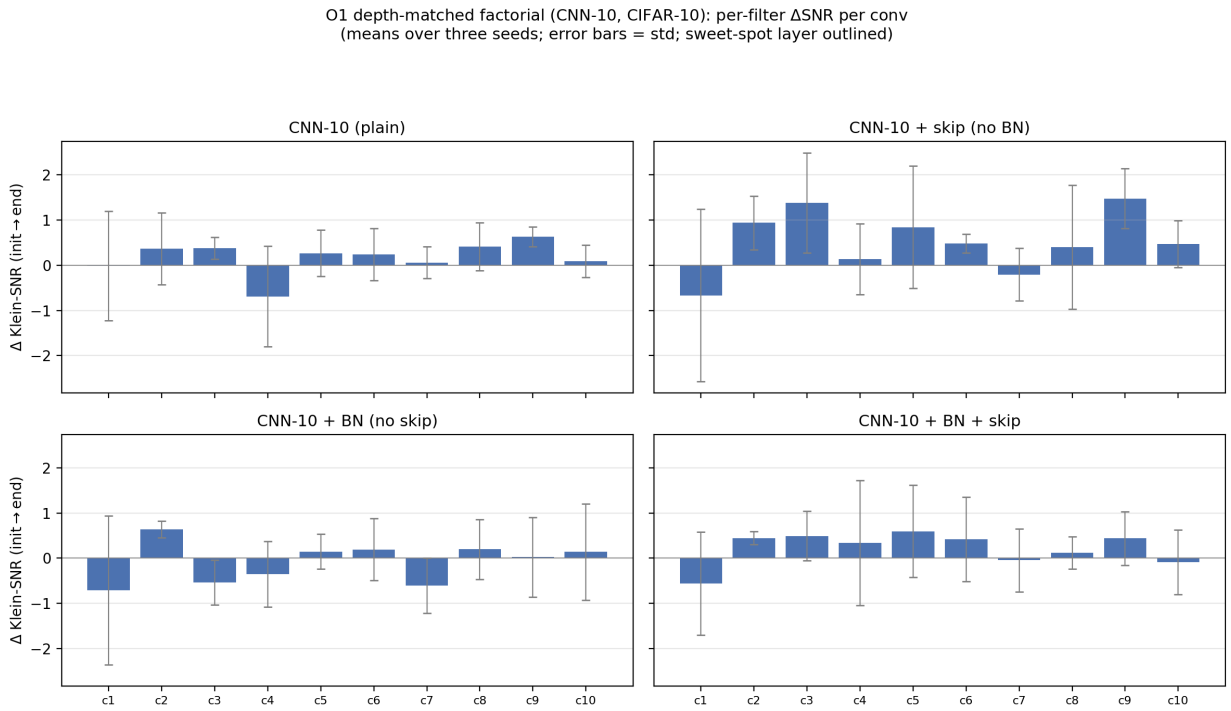

Figure 12: The depth-10 2×2 factorial under the common lr-0.01 recipe: per-filter ΔSNR per convolution for the four cells (means over three seeds; error bars are std). No layer of any cell exceeds +1.5 on average; no bar meets the +3 sweet-spot threshold used throughout, so none is outlined. The signal returns at lr 0.05 in the BN-bearing cells (Figure 13).

the degree of fitting, or of dimensional compression, all of which persist; it is a property of the *high-learning-rate regime*, in the spirit of the lazy-versus-rich training distinction (Chizat et al., 2019): with small steps the filters reach a good minimum near their initialisation without reorganising onto the shared Klein-bottle manifold, with large steps they reorganise. This reading makes a testable prediction: batch normalisation, whose practical role is precisely to make high learning rates usable, should restore the phenomenon at depth. Retraining the two BN-bearing cells at lr 0.05, the rate at which the plain and skip-only cells cannot train at all, confirms it in graded form (Figure 13): topological growth returns, strongest in the BN+skip cell, whose deep-stage `conv8` reaches $\Delta\text{SNR} = +5.7 \pm 2.3$ (the per-seed peak sits at `conv8` in two seeds of three), on par with ResNet-18's sweet spot ($+5.93 \pm 1.52$), and weaker, de-localised growth under BN alone (largest mean $+2.1 \pm 1.3$). Even with BN, skip connections and high learning rate, the depth-10 cell organises *late*, not at the first post-stem layer as in ResNet-18, so the early relocation of O1 is not reproduced by BN+skip+depth alone; ResNet's stem and width-doubling stage structure contribute to *where* the organisation settles.

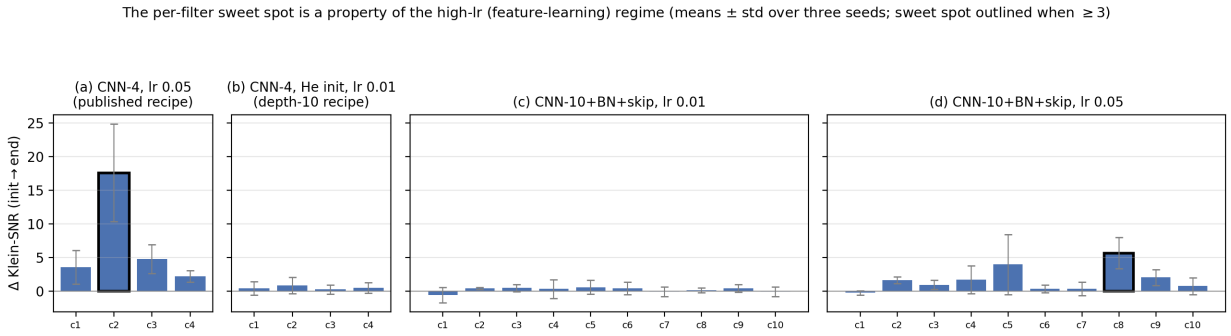

Figure 13: The per-filter sweet spot is a property of the high-learning-rate (feature-learning) regime. **(a)** CNN-4 under the published recipe (lr 0.05) forms its `conv2` sweet spot. **(b)** The same CNN-4 under the depth-10 recipe (He init, lr 0.01) loses it entirely, while fitting *more* (train accuracy 0.96 vs. 0.83). **(c)** The depth-10 BN+skip cell at lr 0.01 is equally flat. **(d)** The same cell retrained at lr 0.05, accessible only to the BN-bearing cells, recovers a deep-stage sweet spot (`conv8`, $+5.7 \pm 2.3$). Means ± std over three seeds; sweet spot outlined when the mean exceeds +3.

D Sweet-spot position across datasets

A natural objection to O1 is that the sweet-spot position might reflect the dataset rather than the architecture, since in the main-text comparison the dataset changes together with the network. To disentangle the two we hold the architecture fixed at CNN-4 and retrain it, with three seeds each, on three further datasets that span a wide range of task complexity at two input geometries: Fashion-MNIST (28×28, single channel, 10 classes), SVHN (32×32, three channels, 10 classes) and CIFAR-100 (32×32, three channels, 100 classes), alongside the original CIFAR-10 (Figure 14).

Across all three 32×32×3 datasets the sweet spot stays firmly at conv2 (CIFAR-10 $+17.6 \pm 7.3$, SVHN $+19.2 \pm 3.1$, CIFAR-100 $+11.3 \pm 2.3$), even though these span a tenfold range of class count and a shift from natural objects to street-view digits. Task complexity therefore does *not* move the sweet spot: it is a property of the architecture, not of how hard the data is. The one case that departs is Fashion-MNIST, whose lower-resolution single-channel inputs flatten the profile so that no single layer dominates (conv2 $+5.8 \pm 1.4$ and conv4 $+7.5 \pm 4.1$ overlap within their seed spread). Its profile, with the weight shifted toward the later convolutions, in fact echoes that of LeNet on MNIST (Figure 2, left panel), the other low-resolution single-channel setting in the paper; the resemblance is not a coincidence but the input-geometry effect showing through two different architectures: on 28×28×1 inputs the early filters have less geometric variety to organise, so the topological organisation crystallises later in depth. We therefore read the sweet-spot position as governed primarily by architecture, with the *input geometry* (resolution and channel count), rather than task complexity, as the secondary modulating factor, a cleaner separation than a single-dataset study would have allowed.

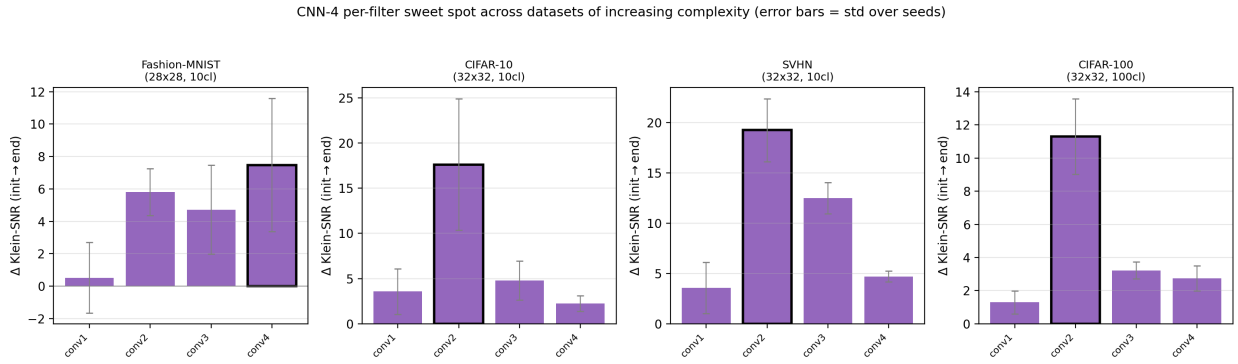

Figure 14: CNN-4 (fixed architecture) per-filter ΔSNR per conv layer on four datasets of increasing complexity, three seeds each (sweet-spot layer outlined). Conv2 remains the sweet spot across all three 32×32×3 datasets (CIFAR-10, SVHN, CIFAR-100) regardless of class count; only the lower-resolution Fashion-MNIST flattens the profile.

E Per-experiment trajectories

The findings in the main text are stated at convergence (init→end deltas) or as peak values. For completeness we give here, for one representative run-family per architecture, the full training-time trajectories of the four descriptors that carry the paper's narrative: the Klein signal-to-noise of the dominant H_1 cycle (Eq. 2), the normalised top-1 H_1 lifetime, the persistence entropy of the H_1 diagram, and the TwoNN intrinsic dimension. In every panel the solid line is the mean over the three seeds $\{7, 42, 123\}$ and the shaded band is ± 1 standard deviation; the horizontal axis is training progress (the optimisation step rescaled to $[0, 1]$) so that families with different step counts are directly comparable. To keep the figures legible we plot a curated subset of layers where a family has many (one 3×3 convolution per stage for ResNet-18; the four layer-0 recurrent gates for the LSTM); all layers are present in the released per-run parquet files. A visible feature of these trajectories is that the checkpoint-to-checkpoint oscillation does not die out late in training, most prominently for the LSTM gates (Fig. 18). This reflects the construction of the descriptors more than residual instability of learning: SNR_1 divides the leading H_1 lifetime by the *median* lifetime of a diagram computed on a small cloud (e.g. 128 rows per LSTM gate), so the appearance or disappearance of a handful of short-lived noise cycles between checkpoints moves the denominator appreciably even when the dominant cycle is stable; the TwoNN estimate on small clouds fluctuates for the same finite-sample reason. The findings in the main text therefore rest on init→end deltas and on trends averaged across seeds, not on per-checkpoint values.

Three features visible in the main results are recovered dynamically here. The CNN-4 `conv2` sweet spot (Fig. 16) emerges as the highest-SNR_1 layer and separates from the others early and monotonically, while its persistence entropy stays low, the signature of a single dominant cycle consolidating. The ViT (Fig. 19) shows the TwoNN dimension of every FFN block contracting steadily through training, with the terminal block's persistence entropy rising late, the decoupling discussed in Sec. 7. The degenerate-free families (MLP, Fig. 15) show the smooth, low-variance trajectories expected of a small trained network.

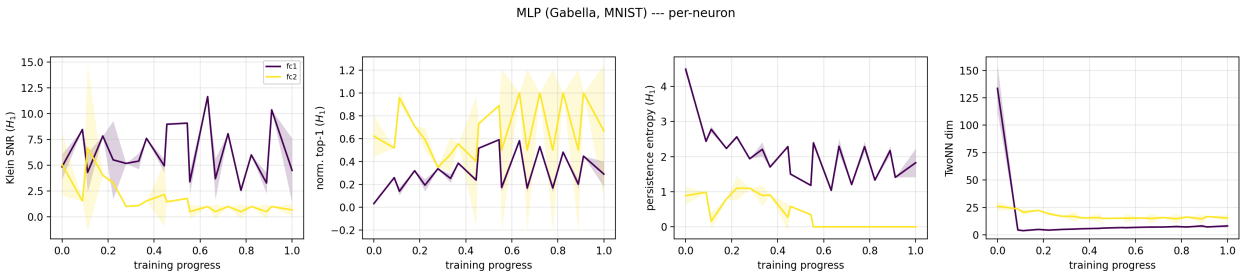

Figure 15: MLP (Gabella reproduction, MNIST), per-neuron point cloud.

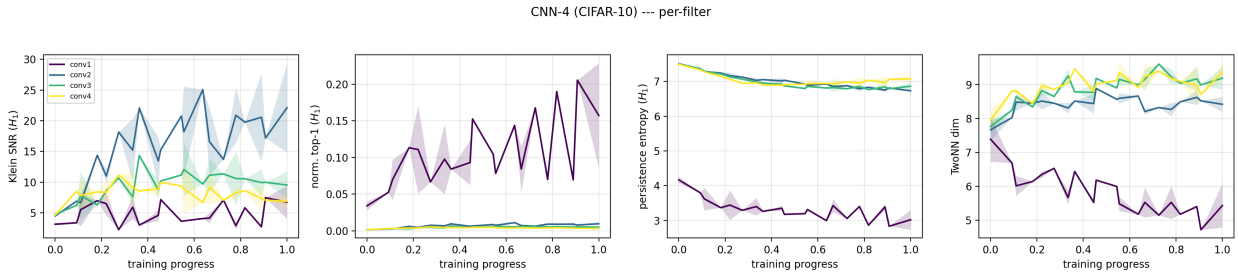

Figure 16: CNN-4 (CIFAR-10), per-filter point cloud. The `conv2` sweet spot dominates SNR_1 and the normalised top-1 lifetime throughout training while keeping the lowest persistence entropy.

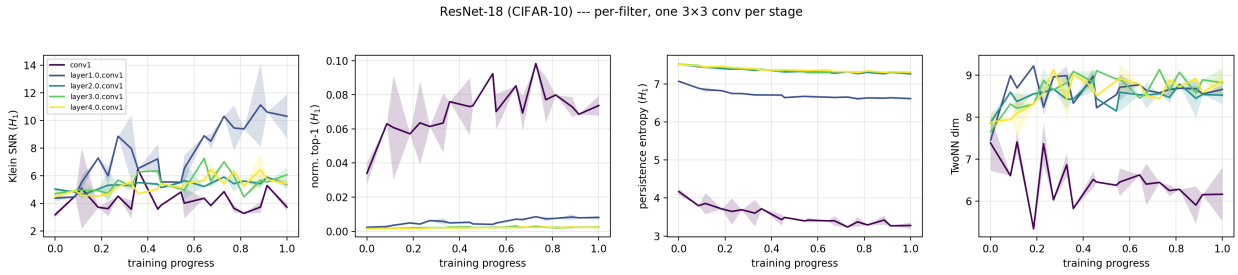

Figure 17: ResNet-18 (CIFAR-10), per-filter point cloud; one representative 3×3 convolution per residual stage.

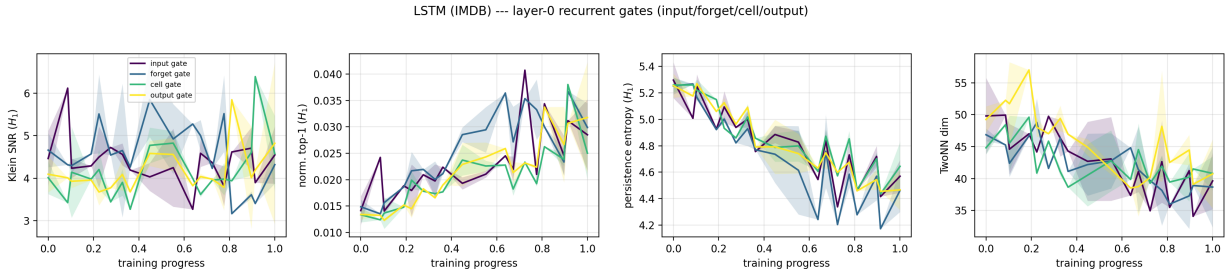

Figure 18: LSTM (IMDB), per-gate point cloud; the four layer-0 recurrent (hidden-to-hidden) gates.

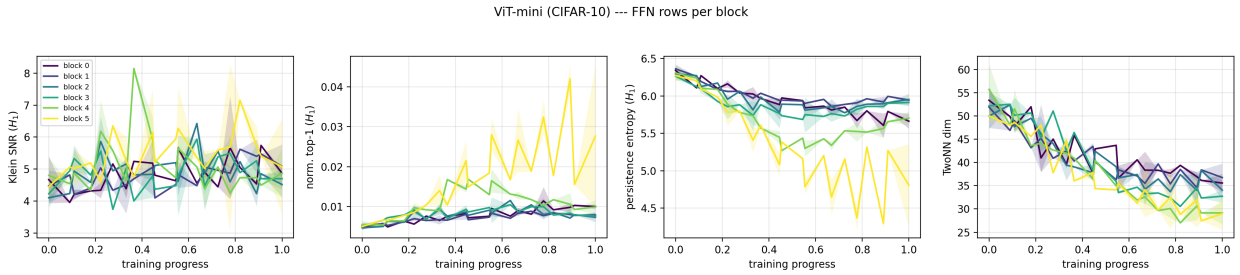

Figure 19: ViT-mini (CIFAR-10), per-FFN-row point cloud, one line per block. The TwoNN dimension contracts in every block; the terminal block's persistence entropy rises late (Sec. 7).

F Held-out topology beyond intrinsic dimension

Intrinsic dimension is a geometric, not a topological, quantity, and its predictive power is close to known activation-dimension results (Ansuini et al., 2019), so a fair question is whether the persistent-homology descriptors contribute anything beyond it. They do. Under the identical leave-one-noise-out protocol (O7) we compare three feature sets, intrinsic dimension alone, persistent homology alone (the H_0/H_1 total persistences, H_1 count, and the Klein summaries), and their union (Figure 20). Adding the persistent-homology descriptors raises the extrapolation R^2 over intrinsic dimension alone in every setting, from 0.62 to 0.82 on CIFAR-10, 0.68 to 0.84 on CIFAR-100, and 0.72 to 0.79 on ResNet-18 (ΔR^2 of +0.20, +0.16, +0.07); persistent homology *alone* matches or exceeds intrinsic dimension alone in all three (0.70, 0.87, 0.75), and on CIFAR-10 a single topological scalar, the H_0 total persistence, already outpredicts the best single intrinsic-dimension descriptor (0.71 vs 0.68). The interpretation is coherent: a memorising network leaves its held-out manifold not only higher-dimensional (intrinsic dimension) but also more *fragmented* into persistent connected components (H_0), two distinct axes of the same degradation that, being distinct, add. We read intrinsic dimension as the best *single* summary but the topological descriptors as carrying independent, additive generalisation signal, not as redundant with it. Given the sample sizes ($n = 15$–25) the magnitudes are indicative; the qualitative result, that persistent homology adds over intrinsic dimension, is consistent across all three settings and permutation-guarded.

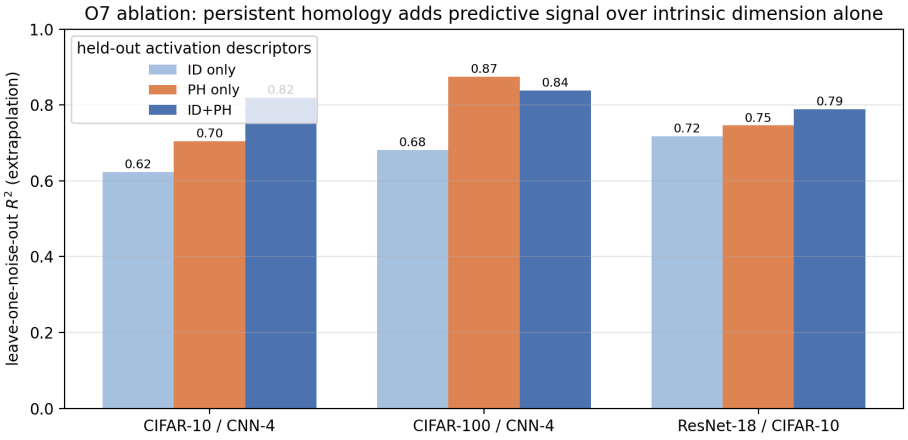

Figure 20: O7 ablation: leave-one-noise-out extrapolation R^2 for three feature sets, held-out intrinsic dimension alone, persistent homology alone, and their union, on the three settings. Persistent homology adds predictive signal over intrinsic dimension alone in every setting and is at least as strong on its own, so the held-out topological descriptors are not redundant with the geometric dimension.

G External validation on real distribution shift (CIFAR-10-C)

This appendix details the CIFAR-10-C validation summarised in Sec. 11. We take a fixed, clean-trained network, CNN-4 (three seeds) and a CIFAR ResNet-18 (two seeds), and evaluate it on CIFAR-10-C (Hendrycks & Dietterich, 2019): 15 corruption types at 5 severities each. For every (corruption, severity) we measure the true accuracy on the labelled corrupted set (ground truth, used only to validate the predictor) and, separately, screen the held-out activation manifold on 512 *unlabelled* corrupted inputs, exactly the screen of Sec. 3. The target is the accuracy drop (clean test accuracy minus corrupted accuracy); the predictor is evaluated under leave-one-corruption-*type*-out cross-validation, so it extrapolates to corruption families never seen in fitting, guarded by a 2000-sample permutation null. Input corruption is a gap-inducing mechanism distinct from the label-noise and train-size sweeps of O6/O7, and was never used to calibrate the predictor.

Table 7 reports the per-layer leave-one-corruption-out Spearman of the held-out H_1 total persistence against the drop. The signal is carried by persistent homology across most layers of both architectures; the early-layer intrinsic dimension (the predictor of the label-noise setting) is flat or sign-inverted (CNN-4 `conv1` $\rho = -0.25$), the same ID-vs-PH dissociation as Fig. 20 but sharper. A predictor fit on three corruptions (`brightness`, `fog`, `gaussian_noise`) confirms on the twelve held-out ones at $R^2 = 0.66$. The locus is architecture-specific: it spans `conv1`–`conv4` in CNN-4 but vanishes at ResNet-18's final block (`layer4`, $\rho = -0.30$), peaking one stage earlier; consequently the numeric zero-shot transfer of the CNN-4 calibration to ResNet-18 fails ($\rho = -0.03$, n.s.), and the descriptor is best used as a within-model ranking. Nominal severity, an oracle unavailable in deployment, scores 0.57 on CNN-4 (below H_1 persistence) and 0.69 on ResNet-18 (above it).

We offer a tentative explanation for the sole failure, ResNet-18's final block ($\rho = -0.30$). It is the only screened representation that is exactly the classifier's input: ResNet applies global average pooling immediately after this block, whereas CNN-4 flattens its final convolution into the head. The pooled, output-adjacent representation is also the most class-collapsed, in the sense of neural collapse (Papyan et al., 2020), so under corruption its held-out manifold contracts toward the class means and its H_1 loop structure, the signal the predictor reads, weakens or inverts. We state this as a hypothesis: we do not measure the collapse directly.

Homology degree (H_2). The screen computes H_0 and H_1 (Vietoris–Rips at `maxdim`$= 1$); the number of candidate simplices for H_2 grows as N^4, and at our sampling density (512 points) 2-cycles are sparse. To check the omission is harmless we recomputed persistence at `maxdim`$= 2$ on a degradation-spanning subset. The H_2 total persistence *does* track the drop, but is redundant with H_1 rather than additive: at `conv3` the two Spearman coefficients are nearly identical (-0.83 vs. -0.84), at `conv4` H_2 is weaker (-0.71 vs. -0.90);

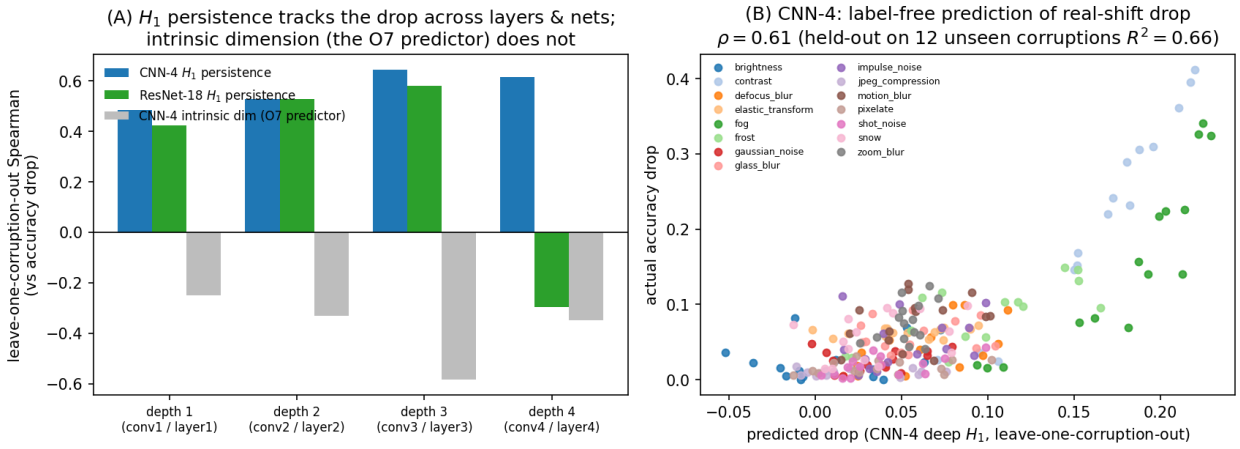

Figure 21: O7 on real distribution shift (CIFAR-10-C). **(a)** Leave-one-corruption-out Spearman of H_1 total persistence vs. the accuracy drop, per depth, for CNN-4 and ResNet-18: the topological descriptor tracks the drop across layers and both architectures, whereas the intrinsic-dimension predictor of the label-noise setting (grey) is flat or inverted; only ResNet-18's final block departs. **(b)** CNN-4 leave-one-corruption-out predicted vs. actual drop from deep H_1 persistence, coloured by corruption family; a predictor formed without these corruptions confirms on the twelve held-out ones ($R^2 = 0.66$), label-free.

Table 7: CIFAR-10-C: leave-one-corruption-out Spearman of held-out H_1 total persistence vs. the accuracy drop, per depth. Persistent homology tracks the drop across layers in both nets except ResNet-18's final block; the early intrinsic-dimension predictor of the label-noise setting does not transfer.

depth	1	2	3	4	nominal severity
CNN-4 (H_1 persistence)	0.48	0.53	0.64	0.61	0.57
ResNet-18 (H_1 persistence)	0.42	0.53	0.58	−0.30	0.69
CNN-4 (intrinsic dim, O7 pred.)	−0.25	flat/inverted			n/a

individual H_2 bars are short-lived (longest $\approx 0.36\times$ the dominant H_1 cycle), i.e. many small voids rather than a robust 2-cycle, at roughly seven times the runtime (8.9 s vs. 1.3 s per cloud). The H_0/H_1 screen therefore loses no distinct signal.