ON THE GEOMETRY OF HIERARCHY: AN EMPIRICAL STUDY IN VISION AND LANGUAGE TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Does hierarchy in the data become hierarchy in the representation, and if so, in what geometric form? The taxonomies that label vision and language datasets are often hierarchical, but a network trained on them is never told that a beagle is a dog, an animal, or a living thing; so whether a model's representations carry the taxonomic tree, and in what shape, must be tested empirically. We define two measurements: whether a child concept's subspace is contained in its parent's, and whether the parent's feature fires concurrently with the child's. In small transformers trained on synthetic data, tree-structured training data gives rise to nested subspaces, whereas data without hierarchical structure does not. Similarly, across all families of pretrained large vision, language, and vision-language models (VLMs) we tested – examining the image encoders of the latter – hierarchical nesting is present, aligns with the ground-truth semantic tree, and varies little with model size. At the same time, the readouts across models differ depending on the training objective. VLMs (e.g., CLIP) preserve the parent firing on the child at the output, ImageNet training with class labels (e.g., ViT) preserve a portion of it, and self-supervised models (e.g., DINO) preserve less of it. Our results indicate that semantic hierarchies in the training data are naturally encoded in nested model subspaces rather than in single directions; however, how much of this hierarchy stays readable differs with the training objective.

1 INTRODUCTION AND RELATED WORK

From Porphyry's tree to WordNet, concepts have long been organized by inheritance: the tree associated with Porphyry's *Isagoge* arranged species beneath genera through successive differences, and WordNet is its modern computational counterpart. In both, a beagle is a dog, a dog is a mammal, and mammals are living things. Taxonomies for both vision and language datasets are constructed this way, yet networks trained on these datasets receive only leaf-level labels or no labels at all. Here we ask two questions: Do the learned representations reflect the dataset's underlying taxonomic structure? If so, what geometric form does that structure take?

We hypothesize that if a taxonomic hierarchy is represented within the network, inheritance should hold in the activations as it holds in the original tree. That is, knowledge about a parent concept should transfer to its subclasses, a probe for the parent should generalize to unseen children, and a feature for the parent should also fire on the child. Although routinely assumed, these properties are rarely measured, which matters for three reasons. *Detection:* A self-supervised or class-supervised encoder is never told that a beagle is a dog, so whether the tree is present at all, where it is instantiated in the network, and whether the answer depends on the training target are open questions. *Method:* A direction gives a linear probe and a steering vector, and a subspace gives a projection, so the geometry that reads a concept is the one along which it can be steered or transferred to its subclasses. *Existing tools*, such as probes and steering vectors (Park et al., 2024; Turner et al., 2023), concept erasure by projection (Ravfogel et al., 2020; Belrose et al., 2023), hierarchy-aware networks (Bilal et al., 2018) and hierarchical sparse autoencoders (Costa et al., 2025), assume a particular geometry for hierarchy, and their effectiveness depends on how well that assumption holds in the network being analyzed.

The latter two reasons rest on what is already known about the geometry of representations. Interpretability has moved through successive pictures of what a representation is, from units (Bau et al., 2017) and directions (Mikolov et al., 2013; Park et al., 2024) to probes (Alain & Bengio, 2017; Belinkov, 2022), to superposition (Elhage et al., 2022) and sparse features (Bricken et al., 2023; Templeton et al., 2024), and to some features being irreducibly multi-dimensional (Engels et al., 2025). The linear representation hypothesis underlying these methods is itself being sharpened, since a feature that is linearly represented need not be linearly accessible, and requiring linear access

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

bounds how many features a layer can hold (Garg et al., 2026). That is, each time something is learned about the geometry, a method follows.

Notably, several tools assume a geometry of hierarchy and build it in. For example, hyperbolic and entailment-cone embeddings place general concepts near the origin and specific ones inside their cones (Nickel & Kiela, 2017; Ganea et al., 2018; Desai et al., 2023; Pal et al., 2025), while order and box embeddings nest the child's box inside the parent's (Vendrov et al., 2016; Vilnis et al., 2018; Li et al., 2019). Matryoshka representations and Matryoshka and matching-pursuit sparse autoencoders put general information in the first coordinates or the first latents (Kusupati et al., 2022; Bussmann et al., 2025; Costa et al., 2025). Hierarchy-aware losses shape the classifier directly (Bertinetto et al., 2020). Finally, sparse autoencoders with an explicit concept hierarchy have been proposed for language models (Muchane et al., 2025). Closest to our question, Park et al. (2025) extend the linear representation hypothesis to hierarchical concepts. In the unembedding space of a language model, whitened so that the causal inner product is Euclidean, they show that the vector of a parent concept is orthogonal to the difference between a child's vector and the parent's, and that the children of a categorical concept form a simplex – something they verify on both WordNet concepts in Gemma and LLaMA. Their result concerns concept vectors, means, and their relative arrangement, whereas our nested subspace hypothesis concerns the the spread of a concept and the directions along which a concept's members vary. The two are complementary: a child–parent mean offset can be orthogonal to the parent's mean vector while still lying within the parent's subspace of variation.

Recent work tests whether unsupervised dictionaries recover hierarchical structure, identifying feature absorption as an obstacle (Grandien et al., 2026). We likewise examine hierarchical structure, but measure it directly in raw activation space rather than whitened unembedding coordinates, testing vision encoders, language models, and VLMs against matched nulls. Our central question is whether, and in what geometric form, hierarchical structure emerges in ordinarily trained networks.

We study this empirically by treating each concept as a cloud of activations elicited by its members and its subspace as the small set of directions along which that cloud varies. We characterize hierarchy through a single geometric relation: the child's subspace is *contained* within the parent's. We call this the *nested subspace hypothesis:* every way in which beagles vary is also a way in which dogs vary, while dogs may vary in additional ways (Rosch et al., 1976)). This relation has a behavioural counterpart: a parent feature *activates* whenever a child feature does. We measure this activation pattern separately using probes or sparse features for the parent.

We test both forms of hierarchy, first in toy transformers trained on data with a controlled hierarchy, then in several families of large vision encoders trained with self-supervision, class supervision, or language supervision, as well as several families of language models, using concepts organized by WordNet. We find that deep networks encode concept hierarchies as nested subspaces and that the training objective determines how much of the hierarchy remains readable at the output. Our contributions are as follows.

- **Controlled evidence that hierarchy in the data produces nested subspaces.** In toy transformers, every child's subspace nests within its parent's across supervision regimes, while training on data without hierarchy produces no nesting (Section 3 and Appendix B).
- **Evidence of nested concept subspaces across large pretrained networks.** Across all vision encoders, language models, and VLMs tested, nesting is present at every depth, follows the semantic tree, strengthens or remains stable toward the output, and varies little with model size (Section 4).
- **Evidence that the training objective shapes how readout of the hierarchy.** At the output, language-supervised vision models retain parent-feature activation for child concepts, class-supervised models retain it partially, and self-supervised encoders lose most of it, with greater losses in larger models. Language models trained solely on text also retain this activation pattern (Section 4).

2 Problem Formulation

All experiments share the same structure. Each network receives inputs whose concepts are organized by a tree, and we examine how a child concept's activations relate to its parent's. The toy, vision, and language settings share three ingredients: definitions of concept activations and subspaces, two

measurements evaluated along parent–child edges, and a matched null for each measurement. Only the data and networks differ between sections.

Concepts and subspaces. A *concept* x is represented by a set of activations $A_x \subset \mathbb{R}^D$, corresponding to its members at a given layer of a given network. We subtract the layer's global mean from all activations once and compute every measurement in raw activation space using the Euclidean inner product – the geometry in which probes, steering vectors, and sparse autoencoders operate. Our measurements are therefore not directly comparable to those reported in whitened unembedding coordinates, as in Park et al., 2025. One running pair is used throughout; for example, $c = \textit{dog}$ and $p = \textit{mammal}$. The subspace of a concept is the PCA basis $B_x \in \mathbb{R}^{D \times d_x}$ of A_x centered at its own mean, holding 90% of the within-concept variance and capped at 80 dimensions. This subspace spans the directions along which the members of x differ from one another, for example, the ways an input can change while remaining a *dog*. However, it says nothing about where the concept sits.

Concept features. The *feature* of a concept x is its difference-of-means detector. The feature direction is the concept mean μ_x after global centering, the difference between the mean of x and the mean of everything, and its threshold sits halfway between the two means,

$$z_x(a) \;=\; \mathrm{ReLU}\big(\langle a, \mu_x/\|\mu_x\|\rangle - \tfrac{1}{2}\|\mu_x\|\big), \tag{1}$$

and x *fires* on a when $z_x(a) > 0$. This is the mass-mean probe of Marks and Tegmark (2024), which generalizes better than a trained logistic probe (Marks and Tegmark, 2024; Li et al., 2023), and the direction along which activation steering moves a model (Rimsky et al., 2024). It requires no training and need not match the detector learned by a trained probe or a sparse autoencoder.

Two measurements. Containment and subsumption are defined as:

$$\mathrm{C}(c \subset p) \;=\; \tfrac{1}{d_c}\,\|B_p^\top B_c\|_F^2 \;=\; \tfrac{1}{d_c}\textstyle\sum_i \cos^2\theta_i, \qquad S(p \mid c) \;=\; P\big(z_p > 0 \mid z_c > 0\big), \tag{2}$$

where θ_i are the principal angles between the two subspaces (the smallest rotations that bring each direction of the child's variation into the parent's span), and B_p is always the held-out basis. For *dog* (child) and *mammal* (parent), $\mathrm{C} = 1$ exactly when the subspace of *dog* lies inside that of *mammal*, and $S(p \mid c)$ is how often *mammal* fires when *dog* does, over a background sample of the layer's activations. Containment is the nested subspace hypothesis itself. Subsumption, what a probe or feature for the parent sees on the child, is a separate measurement and not a consequence of containment. Containment concerns the directions along which a concept varies, subsumption where the child's members sit along the parent's mean direction, and either can hold without the other. Both follow if the child's cloud lies inside the parent's, the picture we test, and tools use both (projections on subspaces, probes on directions). Since a subspace is the linear span of a cloud, span containment is necessary for cloud containment, and curvature inside the spans is a separate question (Appendix E). Base rates force the direction of the subsumption asymmetry, since $P(p|c) - P(c|p) = P(c \cap p)(f_p - f_c)/(f_c f_p) > 0$ whenever parents fire more often (f is a firing rate), so its falsifiable content is that $P(\text{parent} \mid \text{child})$ nears its ceiling and collapses for a matched unrelated concept.

To isolate each measurement from its specific confound, we evaluate it against a null baseline, following the control-task framework of Hewitt & Liang (2019). Specifically, containment is compared to a random subspace matching the child's dimension; subsumption is compared to a control concept that is neither an ancestor nor a descendant of the child, matched to the parent's firing rate; and containment under a shuffled pairing of the same concepts (pairing the child with the held-out subspace of an unrelated parent) is subtracted to isolate the *tree-specific* component. We consider an effect *alive* in a network if its held-out value exceeds the matched null with non-overlapping 95% cluster-bootstrap intervals across parent concepts. Appendix A provides the rationale for each null baseline, exact sample sizes, and a sensitivity sweep.

Held-out parents. Because a parent's members include its child's, the parent's covariance has a range that contains the child's. A parent's subspace estimated from all of its members therefore contains the child's directions by construction, subject to basis truncation. To avoid this artifact, we always estimate the parent's subspace and feature using only members of the parent that do not belong to the child under evaluation. Containment thus tests whether a subspace estimated from siblings captures the held-out child's variation – that is, whether the parent's span generalizes to a child excluded from its estimation. We report containment estimated from all parent members only once, to quanitify the size the artefact (Figure 2), and exclude it from all other analyses.

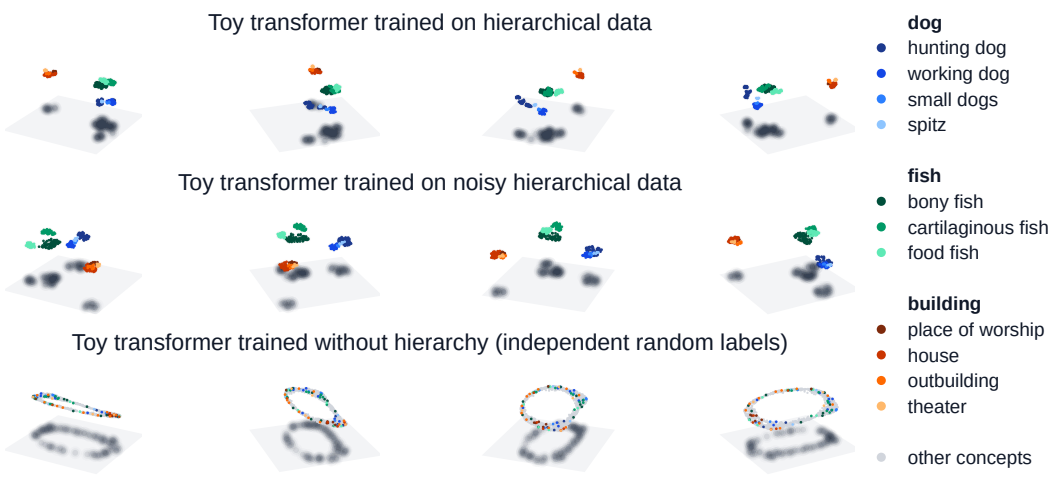

Figure 1: **Toy transformers with and without hierarchy.** Three-dimensional UMAP of the final-layer activations of inputs from the *dog* (blues), *fish* (greens) and *building* (oranges) families. Each point is one input, coloured by its child concept as a tint of the family hue. The top two toys were trained on hierarchical data, the bottom one on independent random labels (other concepts in grey). Each row rotates the same embedding.

3 Toy Models

We start with toy transformers trained on data whose hierarchy we control, and ask whether a transformer trained on hierarchical data acquires nested subspaces, whether one trained without hierarchy does not, and how the nesting changes across layers.

Setup. Inputs are generated from an explicit tree, $X = ZW + \varepsilon$, where Z is the ancestor-closure indicator of a WordNet subtree and W a random projection, so that the hierarchy is fully controlled. We train toy transformers (width 256, 8 to 40 layers, with a width and depth sweep from 64 to 1536 dimensions and 2 to 60 layers) on these inputs, a forty-layer model on the full tree and five eight-layer variants supervised with leaf labels only, parent labels, the full ancestor closure, a noisy variant, and independent random labels as the no-hierarchy control. The same three families (*dog*, *fish*, *building*) are used in every UMAP figure.

Hierarchy in the data produces nested subspaces. We measure held-out containment and subsumption on the toy transformers trained on synthetic conditions. Across all conditions, models trained on hierarchical data consistently nest child concepts within their parent's subspace, whereas the model trained on independent labels does not. In the embeddings (Figure 1) the tree-trained toy models place each family in its own cluster and each child in its own region inside that cluster, whereas the

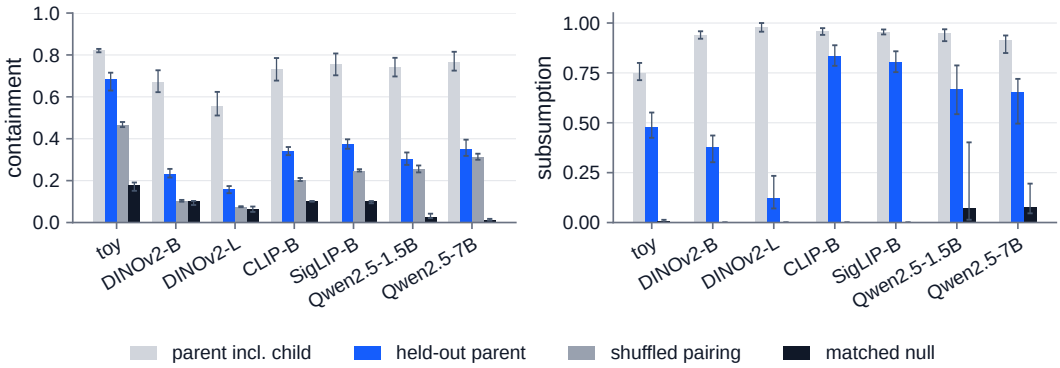

Figure 2: **Held-out parents contain their children.**Containment (left) and subsumption (right) with the parent estimated from all its members (light grey) or from its other children only (blue), in an unrelated parent's held-out subspace (mid grey, shuffled pairing) and under the matched null (dark).

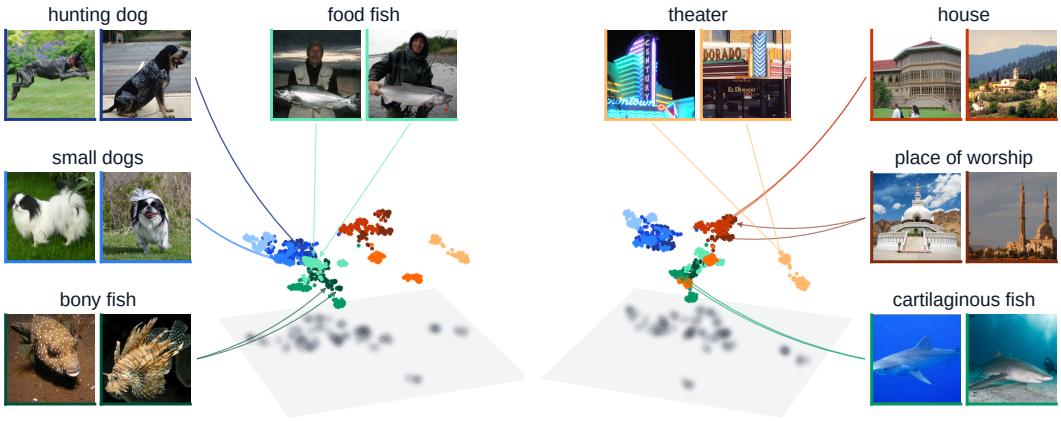

Figure 3: **In DINOv2, each family forms a cluster and each subfamily a region inside it.** Two rotations of one three-dimensional UMAP of the final-layer activations of images from the *dog* (blues), *fish* (greens) and *building* (oranges) families. Each point is one image, coloured by its child concept as a tint of the family hue. Around the panels, eight subfamilies with two example images each, framed in the subfamily's colour, and an arrow from every image to the point it produced.

random-label control is a structureless ring in which the same children are scattered among all other concepts. The numbers agree: held-out containment sits far above the dimension-matched null under every scheme with a tree in the data, including leaf labels only, where the model is never shown any ancestor, and the random-label control's containment is pure set inclusion with no tree-specific part (Appendix A). Explicit ancestor supervision barely changes the geometry, and neither does capacity, since the width and depth sweep (not shown) leaves the nesting where it is. The tree in the data suffices.

Tree-specific nesting grows with depth. We follow both signatures at different depths of the forty-layer model trained on the full tree, each time against the shuffled pairing and the rate-matched null. The raw containment number hides the trend and the tree-specific part shows it. Raw held-out containment is nearly flat from first stored layer to output, but its value under a shuffled pairing falls with depth, so the part that belongs to the tree grows layer by layer and is largest at the output. The rate-matched null of subsumption collapses from the middle of the network onward, and the held-out value stays above it, though it drops from near its ceiling to about half at the output. With both measurements behaving as expected under a controlled hierarchy, we now turn to networks trained on natural images and text.

4 Large Vision and Language Models

We apply the two measurements to large networks and ask how they behave across layers, training targets, and model size. The *training target* of a network is the quantity its loss fits, a text embedding for the language-supervised encoders, a class label for the class-supervised ones, the network's own predictions for the self-supervised ones, and the next token for the language models. We first look at the geometry directly, then test both signatures at the output, and then vary one factor at a time.

Setup. In vision, the 50,000 ImageNet validation images are organized by the WordNet hierarchy (Miller, 1995). A concept is any WordNet node and its members are all images of classes below it (about 1,800 concepts with at least 30 images, and about 1,800 parent-child edges). The vision encoders come in size ladders inside each training target. The self-supervised ladder is DINOv2 from small to giant (Oquab et al., 2024). The class-supervised ladder is the AugReg ViTs trained with one recipe from tiny to large (Steiner et al., 2022), together with ViT base and large (Dosovitskiy et al., 2021). The language-supervised ladders are OpenCLIP trained on one dataset from B/32 to g/14 (Cherti et al., 2023), OpenAI CLIP from B/32 to L/14 (Radford et al., 2021), and SigLIP (Zhai et al., 2023) with its 2025 successor SigLIP 2 (Tschannen et al., 2025). Twenty encoders in all, most at five layers. In language, for about 1,200 WordNet concepts we take up to 30 Wikipedia sentences in which one of the concept's words occurs, with the activation taken as the mean over the word's tokens. Here a parent's members are not the union of its children's (a sentence about mammals is not

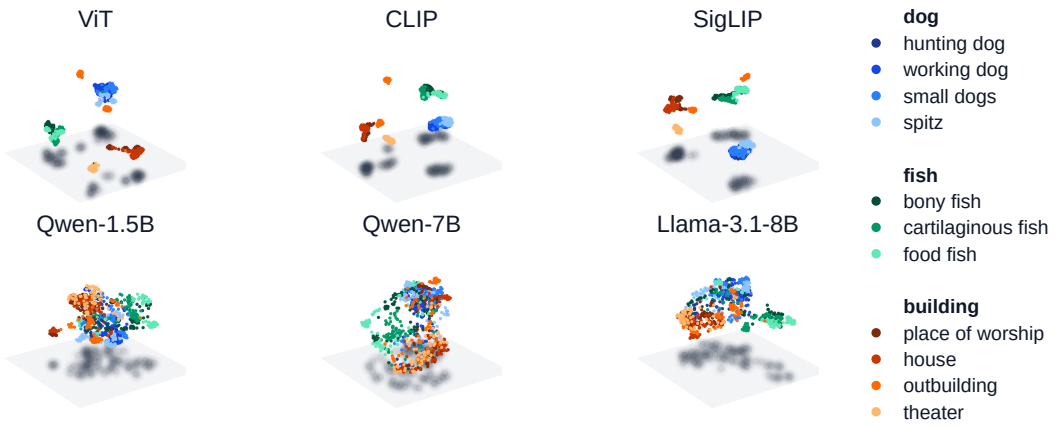

Figure 4: **The same three families in six more networks.** The activation embedding of Figure 3 for three vision encoders (top) and three language models (bottom, one point per sentence occurrence of a concept word), one view each.

a sentence about dogs), which makes the language setting the non-trivial one. The language models are the Pythia suite from 70M to 6.9B, trained with one recipe (Biderman et al., 2023), Qwen2.5 from 0.5B to 7B (Yang et al., 2024), GPT-2 from small to XL (Radford et al., 2019), Qwen 1.5B and 7B (Bai et al., 2023), Llama-3.1-8B (Grattafiori et al., 2024), Gemma-2-2B (Gemma Team, 2024) and Gemma 3 from 1B to 12B (Gemma Team, 2025). Additional details of the coverage of each analysis are listed in Appendix A.1 (Table 3).

Clusters within clusters. Before measuring anything we look at the activations directly. In every network a family of concepts forms a cluster, and inside it each child concept occupies its own region, so the picture is one of clusters within clusters. The nesting is sharpest in the vision encoders, and looser in the language models, whose word-occurrence clouds interleave. Figure 3 shows a representative UMAP of the final-layer activations of DINOv2 for three families, dog, fish and building, with their children, and Figure 4 the same three families in three more vision encoders and three language models. The measurements that follow cover every concept and every parent-child edge of the tree, each read against its null (Tables 1, 2).

Both signatures are present at the output. For 250 parent-child edges per network and layer we compute held-out containment against the dimension-matched random subspace and against the shuffled pairing, and held-out subsumption against the rate-matched unrelated concept (Table 1 for the vision encoders and Table 2 for the language models). Every large network nests its concept subspaces, and in every vision encoder and in most language models the parent's feature fires on the child far more often than an unrelated concept of the same frequency does, the exceptions being residual streams whose rate-matched null is itself large (Table 1). In every vision encoder held-out containment exceeds its random null with intervals clear of it, and a child sits measurably deeper inside its own parent than inside a random other concept, with intervals clear of zero in every encoder. Subsumption shows the texture the hypothesis predicts. Parents fire more often than their children, the child's own direction explains more of a child instance's norm than the parent's does, and if the child fires the parent fires. The tree that is nested is the semantic one. An independent biological taxonomy over the same images, the NCBI classification matched by Latin binomials, reads at its null while WordNet on the same species reads well above it (Appendix B). The table also shows the pattern the rest of this section explains. Held-out containment is of similar size in every vision encoder, while the subsumption readout ranges from nearly absent to nearly complete. The next three paragraphs vary depth, training target and size to see what moves each of the two.

Effect of depth. We repeat the containment measurement at five relative depths of every encoder, each time against the random subspace and against the shuffled pairing, so that the raw number splits into a generic part (any concept inside any other) and a tree-specific part (the child's own parent minus a random other concept). The tree-specific part survives to the output in every encoder, while the generic overlap between concepts fades. Read naively, nesting is high in early layers and falls toward the output in every vision model. The shuffled pairing shows that this decline is almost

network	training target	year	held-out containment	subsumption $P(\text{parent} \mid \text{child})$
DINOv2 B	self-supervised	2023	0.23	0.38
ViT B	class-supervised	2021	0.23	0.67
AugReg ViT B	class-supervised	2022	0.20	0.65
CLIP B/16	language-supervised	2021	0.34	0.83
OpenCLIP B/32	language-supervised	2023	0.33	0.84
SigLIP B/16	language-supervised	2023	0.38	0.80
SigLIP 2 B/16	language-supervised	2025	0.40	0.80

Table 1: **Both measurements at the output of seven vision encoders, one per family.** Held-out containment and subsumption, ordered by year inside each training target, with cells shaded in the target's colour in proportion to their value.

network	year	held-out containment	subsumption $P(\text{parent} \mid \text{child})$
GPT-2-large	2019	0.21	0.37
Pythia-410M	2023	0.26	0.47
Qwen-1.5B	2023	0.30	0.67
Qwen2.5-1.5B	2024	0.30	0.69
Llama-3.1-8B	2024	0.21	0.44
Gemma-2-2B	2024	0.33	0.34
Gemma-3-1B	2025	0.24	0.37

Table 2: **The same measurements at the output of seven language models, one per family.** Ordered by year, shaded in proportion to their value.

entirely generic. Early in a network any concept's subspace sits inside any other's, and it is this shared overlap that fades with depth. The tree-specific part behaves in the opposite way, as it did in the toy (Figure 5, left). It grows from the first blocks through the second half of every network and remains at the output in every encoder profiled in the figure. Among them, the language-supervised and class-supervised encoders rise to the output, and the self-supervised DINOv2 family peaks at about three-quarters depth and eases off, as do the AugReg ViTs and SigLIP 2 outside the figure. The geometry is therefore not discarded with depth. What changes is how much of the overlap between concepts is generic and how much is the tree. Whether the parent's feature still fires when the child's does at the output turns out to depend on the training target, which we examine next.

Effect of language. We compare the three training targets at the final layer on the readout of the geometry, held-out subsumption (does the parent's feature still fire when the child's does?), and we run paired per-edge tests between every language-supervised and every non-language encoder on identical WordNet edges (Table 1, and Figure 7 in Appendix D). Language supervision keeps the parent firing on the child at the output, class labels keep part of that readout, and self-supervision loses most of it. The three language-supervised encoders of the paired test (CLIP, SigLIP and OpenCLIP, from different groups, data and losses) sit at the top, the class-supervised ViTs in the middle, and the self-supervised DINOv2 family at the bottom, after peaking mid-network. On identical WordNet edges every paired comparison between a language-supervised and a non-language encoder is positive, on subsumption and on containment alike. One qualification follows from the shuffled-pairing control. The containment part of that gap is mostly generic overlap, which is higher in the language-supervised outputs, and the tree-specific edge is small. Text alignment therefore pushes the output toward subspaces that are readable, and its effect is an effect on the readout more than on the nesting itself. The language-supervised models differ from the others in training data as well as in objective, so we describe the finding as an effect of language-supervised training, the objective-plus-data package (Muttenthaler et al., 2023). The existing models rule out the simplest data-side explanation, since DINOv2 is trained on web-scale curated images without language and loses the readout nevertheless.

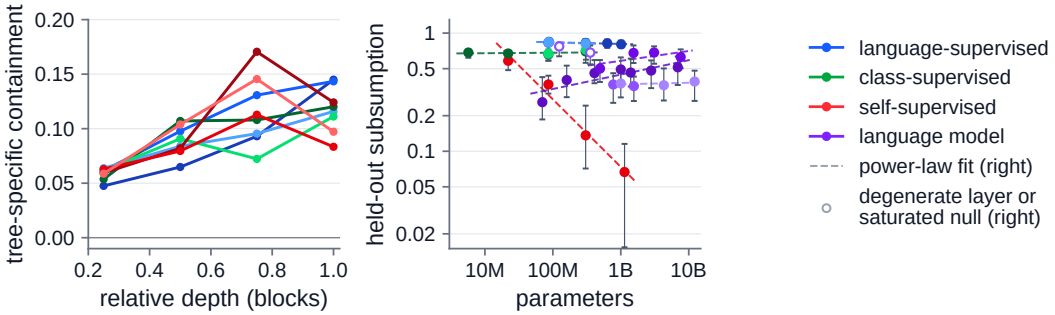

Figure 5: **Tree-specific nesting grows with depth, and only DINOv2 loses its readout with size.** (Left) Tree-specific containment across relative depth in the vision encoders, one line per encoder. (Right) Held-out subsumption at the output against parameter count on logarithmic axes, one set of points and one dashed power-law fit per family. Colours mark the training target and tones the models or families within it. Hollow markers are degenerate layers or saturated nulls.

Effect of scale. Within each training target we follow a ladder of sizes inside one family (for example DINOv2 from small to giant, the ViTs from tiny to large, or OpenCLIP on one dataset from B/32 to g/14), together with the two ViT sizes, and we fit a power law in the parameter count to tree-specific containment and to held-out subsumption at the output, with a cluster bootstrap over parents and a paired comparison of the largest and the smallest model on the same edges (Figure 5, right, and Appendix D). Size changes the nesting little and changes the readout only where the training target lets it. In the language models, tree-specific containment is flat or nearly so over two orders of magnitude, with Gemma 3 the one family whose fitted decline is clear of zero. In the vision encoders it eases slowly at the output as models grow, by roughly a fifth to a third for ten times the parameters, and at a matched relative depth the decline disappears in DINOv2 and in both CLIP families, so it is largely an effect of the larger models being deeper, while the AugReg ladder keeps its decline at the output and at the matched depth. Held-out subsumption falls steeply with size under self-supervision, by an order of magnitude from DINOv2 small to giant, and the small model reads out best. It holds under class labels, under language supervision and in Gemma 3, and it rises slightly with size in Pythia and Qwen2.5. At a matched relative depth the self-supervised readout is flat, so the larger DINOv2 models lose it in their final layers rather than never having it. The tree rebuilt from activations is as good in the largest model of a ladder as in the smallest (Appendix D). Scale therefore neither creates the geometry nor removes it. It amplifies what the training target does to the readout, in line with the finding that objective and data rather than scale drive the human alignment of vision representations (Muttenthaler et al., 2023). We now check the language models more closely, where the same reasoning makes a prediction.

Language models carry the same two signatures. We measure containment at every stored layer and subsumption at five relative depths on the twenty-two language models in Appendix A.1. Because residual streams are dominated by high-variance directions, we score containment against the shuffled pairing as well as against the isotropic random subspace, and repeat all analyses with each dimension divided by its standard deviation. Both signatures are present, but the isotropic null substantially overstates tree-specific containment. Against a dimension-matched random subspace, held-out containment appears large. That null is isotropic and residual streams are not, so a child sits inside a random other concept's subspace almost as well as inside its parent's. Against the shuffled pairing the tree-specific part is real, with intervals clear of zero in most models, but an order of magnitude smaller (Table 2). In the most anisotropic final layer, GPT-2-medium, any concept contains any other and nothing can be read. Subsumption is the more robust signature here and stays above its rate-matched null in every model whose null does not saturate, and clear of it in most. That rise is the shuffled baseline rising as final layers become anisotropic. Tree-specific containment is a few hundredths at most. It is nil in the early and middle layers of GPT-2, most Pythia sizes and Qwen2.5, and flat at that level through Llama, Gemma and the larger Pythia models. We keep the correction in the main text because the isotropic null is the field's default. Language models thus behave as a text target predicts. They nest, weakly but specifically, and they keep the parent firing on the child.

The geometry of hierarchy. Given only the activations, the two signatures rebuild most of the WordNet tree, orienting nearly every true edge where hierarchical clustering cannot tell parent from

child (Appendix B). A linear probe trained to identify a parent concept also recognizes unseen subclasses, achieving more than ten times chance accuracy in vision encoders and approximately four times chance accuracy in GPT-2 (Appendix B).

5 Discussion and Conclusion

We empirically studied representation hierarchy across toy transformers, vision encoders under three objectives, and language models using two metrics. Across architectures, child concepts consistently vary along directions spanned by their parents, and parent features activate on child inputs. This nesting reflects semantic rather than biological taxonomies, persists or strengthens toward output, and remains stable across scale. It forms nested linear spans whose concept-cloud curvature carries no additional hierarchy. Other proposed geometric structures—boxes and order embeddings (Vendrov et al., 2016; Vilnis et al., 2018), entailment cones (Ganea et al., 2018), and coordinate prefixes (Kusupati et al., 2022)—fail the controls, while norm-as-depth (Nickel & Kiela, 2017) survives only as a coarse heuristic (Appendix B, Appendix E).

The networks differed in how much of this structure their output keeps readable. Language-supervised encoders keep the parent's feature firing on the child, class-supervised encoders keep part of it, and self-supervised encoders lose most of it, more so in larger models. Language models keep both a weak but tree-specific nesting and the parent's firing on the child. We hypothesize that the training target explains these differences. A text embedding or a next-token distribution carries a hierarchy of its own, since the contexts of *beagle* are a subset of the contexts of *dog* (Geffet & Dagan, 2005). Concurrent recent work shows that hierarchical geometry can emerge from such co-occurrence statistics alone (Nava & Wyart, 2026), so fitting these targets may keep the parent readable. One-hot labels carry no such structure, and cross-entropy over mutually exclusive classes drives class means toward an equiangular frame in which every pair of classes is equally far apart (Papyan et al., 2020). Self-supervised objectives reward spreading instances apart, through the uniformity term of contrastive learning (Wang & Isola, 2020) or the regularizer that spreads the features of DINO within a batch (Oquab et al., 2024), and none of them has a target that carries the hierarchy. The hypothesis echoes psychology, where verbal labels organize perceptual categories (Lupyan, 2012; Lupyan & Ward, 2013; Boutonnet & Lupyan, 2015; Suffill et al., 2024), and neuroscience, where language supervision yields better models of high-level visual cortex (Wang et al., 2023). A direct test, adding captions or a hierarchy-aware loss (Bertinetto et al., 2020) to a class-supervised encoder, remains future work.

Two practical implications follow. First, because hierarchy resides in concept variation spans, probes (Alain & Bengio, 2017), steering vectors (Turner et al., 2023; Zou et al., 2023), and featurizers (Bricken et al., 2023) should target nested spans rather than rigid box or prefix geometries. Second, evaluating subspace containment in residual streams requires shuffled parent–child controls alongside isotropic nulls to avoid overstating hierarchy. Appendix C takes a first step toward building this geometry into sparse autoencoders, with a coarse-to-fine dictionary that makes the tree readable from its latents.

Subspace nesting also suggests safety monitors could operate at the category level rather than cataloging single inputs, as internal monitors (Burns et al., 2023; Zou et al., 2023) can generalize to unseen subclasses. Still, caution is warranted because data-driven semantic structures may allow atypical harms to evade family monitors. Moreover, steering along a parent's direction was no more specific to child concepts than steering along unrelated directions (Appendix A), meaning family-level oversight must be validated on held-out subcategories.

Limitations. Shared concept samples mean model evaluations are independent, but concept draws are not. The semantic-versus-biological comparison covers roughly 100 species matched between WordNet and NCBI. Scaling ladders span three to seven models per family, and only Pythia and AugReg ViTs fix both data and recipe (OpenCLIP fixes only data), making scaling trends descriptive rather than causal. Language-model final layers are anisotropic, so containment is reported strictly as tree-specific excess over matched controls. Finally, our metrics characterize linear spans; internal non-linear geometry is probed only via the curvature tests in Appendix E.

AI USE STATEMENT

Generative AI tools were used to assist with coding and to proofread the paper. All code and text were verified by the authors, who take responsibility for the final content of this work.

REFERENCES

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *ICLR Workshop*, 2017.

J. Bai et al. Qwen technical report. *arXiv:2309.16609*, 2023.

D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. *CVPR*, 2017.

Y. Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

N. Belrose, D. Schneider-Joseph, S. Ravfogel, R. Cotterell, E. Raff, and S. Biderman. LEACE: Perfect linear concept erasure in closed form. *NeurIPS*, 2023.

L. Bertinetto, R. Mueller, K. Tertikas, S. Samangooei, and N. A. Lord. Making better mistakes: Leveraging class hierarchies with deep networks. *CVPR*, 2020.

S. Biderman et al. Pythia: A suite for analyzing large language models across training and scaling. *ICML*, 2023.

A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–162, 2018.

B. Boutonnet and G. Lupyan. Words jump-start vision: A label advantage in object recognition. *Journal of Neuroscience*, 35(25):9329–9335, 2015.

T. Bricken et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. *ICLR*, 2023.

B. Bussmann, N. Nabeshima, A. Karvonen, and N. Nanda. Learning multi-level features with matryoshka sparse autoencoders. *ICML*, 2025.

D. Chanin, J. Wilken-Smith, T. Dulka, H. Bhatnagar, and J. Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv:2409.14507*, 2024.

M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev. Reproducible scaling laws for contrastive language-image learning. *CVPR*, 2023.

U. Cohen, S. Chung, D. D. Lee, and H. Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11:746, 2020.

V. Costa, T. Fel, E. S. Lubana, B. Tolooshams, and D. Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv:2506.03093*, 2025.

K. Desai, M. Nickel, T. Rajpurohit, J. Johnson, and R. Vedantam. Hyperbolic image-text representations. *ICML*, 2023.

A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

N. Elhage et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

J. Engels, E. J. Michaud, I. Liao, W. Gurnee, and M. Tegmark. Not all language model features are one-dimensionally linear. *ICLR*, 2025.

O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. *ICML*, 2018.

N. Garg, J. Kleinberg, and K. Peng. How many features can a language model store under the linear representation hypothesis? *arXiv preprint arXiv:2602.11246*, 2026.

M. Geffet and I. Dagan. The distributional inclusion hypotheses and lexical entailment. *ACL*, 2005.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv:2503.19786*, 2025.

N. Grandien, D. Steinmann, F. Friedrich, and K. Kersting. Do sparse autoencoders learn meaningful concept hierarchies? *arXiv preprint arXiv:2606.22994*, 2026.

A. Grattafiori et al. The Llama 3 herd of models. *arXiv:2407.21783*, 2024.

J. Hewitt and P. Liang. Designing and interpreting probes with control tasks. *EMNLP*, 2019.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. *ICML*, 2010.

L. Kotlerman, I. Dagan, I. Szpektor, and M. Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389, 2010.

A. Kusupati, M. Wallingford, V. Ramanujan, R. Somani, J. S. Park, K. Pillutla, P. Jain, S. Kakade, and A. Farhadi. LLC: Accurate, multi-purpose learnt low-dimensional binary codes. *NeurIPS*, 2021.

A. Kusupati et al. Matryoshka representation learning. *NeurIPS*, 2022.

X. Li, L. Vilnis, D. Zhang, M. Boratko, and A. McCallum. Smoothing the geometry of probabilistic box embeddings. *ICLR*, 2019.

K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *NeurIPS*, 2023.

G. Lupyan. Linguistically modulated perception and cognition: The label-feedback hypothesis. *Frontiers in Psychology*, 3:54, 2012.

G. Lupyan and E. J. Ward. Language can boost otherwise unseen objects into visual awareness. *PNAS*, 110(35):14196–14201, 2013.

S. Marks and M. Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *COLM*, 2024.

T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. *NAACL*, 2013.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *ICLR*, 2017.

G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

M. Muchane, S. Richardson, K. Park, and V. Veitch. Incorporating hierarchical semantics in sparse autoencoder architectures. *arXiv preprint arXiv:2506.01197*, 2025.

L. Muttenthaler, J. Dippel, L. Linhardt, R. A. Vandermeulen, and S. Kornblith. Human alignment of neural network representations. *ICLR*, 2023.

A. Nava and M. Wyart. Hierarchical concept geometry in language models emerges from word co-occurrence. *arXiv:2605.23821*, 2026.

M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. *NeurIPS*, 2017.

M. Oquab et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

A. Pal, M. van Spengler, G. M. D'Amely di Melendugno, A. Flaborea, F. Galasso, and P. Mettes. Compositional entailment learning for hyperbolic vision-language models. *ICLR*, 2025.

V. Papyan, X. Y. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *PNAS*, 117(40):24652–24663, 2020.

K. Park, Y. J. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *ICML*, 2024.

K. Park, Y. J. Choe, Y. Jiang, and V. Veitch. The geometry of categorical and hierarchical concepts in large language models. *ICLR*, 2025.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI technical report*, 2019.

A. Radford et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, and Y. Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. *ACL*, 2020.

N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner. Steering Llama 2 via contrastive activation addition. *ACL*, 2024.

E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8, 382–439.

M. Sanderson and B. Croft. Deriving concept hierarchies from text. *SIGIR*, 1999.

A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. *TMLR*, 2022.

E. Suffill, J. van Paridon, and G. Lupyan. Mind melds: Verbal labels induce greater representational alignment. *Open Mind*, 8:950–971, 2024.

A. Templeton et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024.

M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv:2502.14786*, 2025.

A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization. *arXiv:2308.10248*, 2023.

I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *ICLR*, 2016.

L. Vilnis, X. Li, S. Murty, and A. McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. *ACL*, 2018.

T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ICML*, 2020.

Wang, A. Y., Kay, K., Naselaris, T., Tarr, M. J., and Wehbe, L. Better models of human high-level visual cortex emerge from natural language supervision with a large and diverse dataset. *Nature Machine Intelligence*, 5(12):1415–1426, 2023.

J. Weeds and D. Weir. A general framework for distributional similarity. *EMNLP*, 2003.

A. Yang et al. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. *ICCV*, 2023.

A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv:2310.01405*, 2023.

A Additional Method Details

Controls in detail. Each measurement must be high if the hypothesis holds, and a reading at its null rules the hypothesis out for that network. The *held-out parent* is not a control but the definition of the measurement. In every real dataset the parent's samples contain the child's, so a parent estimated from all of its members contains the child by construction, and we therefore estimate the parent from its other children only, which turns containment into a question of generalization, whether the parent's span recovers the variation of a child it was not built from. The results in this paper are held out unless saying the opposite, and the held-out statement is the stronger one, since a parent subspace built from some children contains a child it has never seen. The *dimension-matched random subspace* control compares every containment with a random subspace of exactly the child's dimension, since two random subspaces already overlap by chance and more so when they are large. The *rate-matched non-ancestor* control compares subsumption with an unrelated concept matched in firing rate to the parent, since a frequent unrelated concept can fire on almost everything. The concept is neither an ancestor nor a descendant of the child, and it is the closest in background firing rate to the parent among ten random candidates. The *shuffled pairing* control subtracts a measurement's value under a shuffled pairing of the same concepts, which leaves its tree-specific content. It is computed for containment. Each child is paired with the held-out subspaces of three other parents of the same network and layer, none of them an ancestor of the child, and the mean of the three is subtracted. Uncertainty is reported with cluster bootstrap over parent concepts, since edges sharing a parent are not independent, with 95% intervals throughout. Per network and layer we sample 250 edges, up to 400 activations per concept, and a background of $20{,}000$ activations for firing rates. The literature has proposed other pictures of hierarchy (depth as norm, entailment cones, boxes, coordinate prefixes). Appendix B derives a measurement for each, shows on synthetic activations that each responds only to its own picture, and finds that none of them survives in trained toys or in real networks, so the main text stays with the two measurements of Section 2.

Language concepts. The language concepts are the nodes of the same ImageNet WordNet tree as in vision. A concept's words are its single-word WordNet lemmas, lower-cased. Multiword lemmas are not used. We split the first 300 million characters of WikiText-103 (Merity et al., 2017) into 2.1 million sentences of 30 to 400 characters. We then search them for exact whole-word matches without lemmatization, so a plural does not match its singular. We keep up to 30 matching sentences per concept, drawn at random, and drop the concepts with fewer than 5. This leaves $1{,}151$ concepts and $31{,}369$ occurrences. No sense filtering is applied. A word shared by several concepts (84 words) counts for each of them. A sentence that uses another sense of a word is also kept, for example *jaguar* in a sentence about the car maker. Each occurrence is a member of the concept whose word it contains and of every ancestor of that concept. A parent's members are therefore its own sentences together with all the sentences of its descendants. The held-out parent is built exactly as in vision. It drops the sentences of the child and of its descendants, and it keeps the parent's own sentences and those of its other children. Unlike an image, a sentence can name the parent itself (a sentence about mammals is not a sentence about dogs). Such sentences are a median of 8% of a held-out parent's members (mean 29%). A concept enters the analysis with at least 30 members ($1{,}177$ concepts). An edge enters when its held-out parent has at least 30 members ($1{,}047$ edges, of which 250 are drawn per layer). A leaf has at most 30 sentences, so a leaf child is estimated from exactly 30 and its subspace has at most 29 dimensions. The shuffled pairing uses the same child subspace, so the noise of this estimate enters the measurement and the control alike. A sentence's activation is the mean over the tokens that overlap the matched word, with sentences truncated to 64 tokens. When no token overlaps the word, the nearest token is read (just over 1% of the occurrences).

Robustness audit. Five claims of our own did *not* survive and are retracted rather than softened. First, "$P(\text{parent} \mid \text{child}) \approx 1$" was an artefact of parents containing their children's samples, and held out it is 0.12 to 0.83 in the networks of Figure 2. Second, "vision nests strongly early" was largely generic early-layer nesting (the shuffled-pairing null rises with it), and tree recovery is best at the output. Third, "Matryoshka is a rival picture of concept hierarchy" failed because its objective produces no concept-hierarchy signature. Fourth, "activation steering along a parent direction is inheritance-specific" failed because a steering-specificity matrix reproduces but non-parent distractors steer just as well, so the effect is generic concept-to-token steering. Fifth, "language models build nesting toward the output" and "vision models discard it" were both the isotropic null misreading anisotropy and generic overlap, and the shuffled-pairing audit (Table 2) replaces them with the tree-specific readings of Section 4.

Three further audits strengthen the surviving claims. A Poincaré-ball pipeline (hyperbolic MDS of the same activation similarities, decoded by norm-ordering or entailment cones, ball dimensions 8 and 32) recovers almost none of the tree (ancestor-F_1 0.02 to 0.05 against 0.58 to 0.88 for containment), so the live rival reduces to the raw norm heuristic. An independent reimplementation of the held-out battery from the paper's text alone (different PCA solver, different subspace routine, different sampling code and seeds) reproduces containment to about 0.01 on three models. And rewiring a fraction p of tree edges to random parents decays held-out containment monotonically, from 0.24 at $p{=}0$ to 0.12 at $p{=}1$, the shuffled floor, so the signature tracks the tree's content and not the machinery.

Sensitivity to analysis hyper-parameters. Varying the PCA variance threshold (0.85 to 0.95), the dimension cap (40 to 120), the per-concept sample budget (200 to 600) and the firing threshold (0.25 to $0.75\times$ margin) one factor at a time changes no conclusion. Held-out containment stays 2.3 to $3.5\times$ its null on DINOv2 and an order of magnitude above it on Qwen, and the firing margin over its rate-matched null persists at every threshold.

Toy data and training. The toy tree is the WordNet noun hierarchy below *entity* ($74{,}374$ concepts), made a tree by keeping each concept's shortest-path parent. Each training row picks a leaf at random. Its row of Z marks that leaf and all of its ancestors (3 to 19 concepts, 9.1 on average). The input is $X = ZW$, where W has 165 columns and independent $\mathcal{N}(0, 1/165)$ entries. There is no noise term except in the noisy variant, which adds Gaussian noise with 0.3 times the standard deviation of X. A linear map turns each input into eight tokens of width 256, and a class token is prepended. Every sub-block is a pre-norm residual branch whose gate starts at 10^{-9}. Every toy is trained on one million rows with a multi-label binary cross-entropy (positive weight capped at 100). The toys differ only in their labels and inputs. The forty-layer toy has twenty blocks, each an attention and a feed-forward sub-block, and is trained for 30 epochs on the full ancestor closure. The eight-layer variants have four blocks and train for at most 20 epochs, stopping once the top-k F_1 reaches 0.9. Their labels are the deepest concept of each row (leaf labels), its two deepest concepts (leaf and parent labels), its four deepest concepts, or the full closure with noisy inputs. The leaf-label variant keeps the tree in the inputs and removes it from the labels, since it never sees an ancestor. The no-hierarchy control changes both. Each of its rows is a random set of concepts of the same size, drawn without the tree. This set is both the input code and the label, so neither the inputs nor the labels carry the tree. We read the class token at every stored layer on $50{,}000$ rows. A stored layer is the residual stream after one attention or feed-forward sub-block. The first stored layer therefore follows one attention sub-block, and the input itself is not stored. At initialization every gate is 10^{-9}, so an untrained network gives every input the same class token, to one part in a billion. The baseline for geometry inherited from the data is therefore the input X itself. In every toy a concept's members are the rows of its descendants, so all toys are measured on the same concepts and edges.

Toy training schemes. Held-out containment (against the dimension-matched null) for toy transformers trained with different supervision is 0.79 (0.02) for leaf labels only, 0.78 (0.04) for parent labels, 0.76 (0.08) for the full ancestor closure, 0.81 (0.09) for the noisy variant and 0.68 (0.18) for the forty-layer model of the main text trained on the full tree. A model trained on independent random labels cannot be evaluated with tree masks, because any two random unions of one distribution trivially contain each other (containment 0.998). The correct null there is the shuffled pairing, and a flat-label toy encoder reads near zero on every signature (Figure 6). This saturation caution is why every containment number in the paper is paired with a matched null.

Multiple parents. For WordNet concepts with several parents (for example *whale* under both *mammal* and *aquatic animal*, about 37 concepts), the child nests robustly in its dominant parent, and a small second-parent signal survives similarity-matched decoys (1.26 to 1.30 times, replicated across four LLM families and seven layers). Polysemy (one word, several senses) shows no subspace signature. Senses are read from context, not stored as nested subspaces. Given the small concept sample, we report both as exploratory.

A.1 Networks and Coverage

The master battery (nested subspaces, subsumption, the prefix, norm, box and tree-distance detectors, all held out with cluster-bootstrap intervals) covers all 48 networks and the shuffled-pairing audit all but two toy variants. The more expensive analyses ran on subsets, always including at least one model of each training target. Each analysis names its networks where it is reported.

Networks	Training target	Width	Layers analysed
toy transformers (six, trained on the full tree, and on leaf-only, parent, ancestor-closure, noisy and independent random labels)	tree labels (synthetic)	256	5 of 40 (full), final (variants)
DINOv2 small / base / large / giant (Oquab et al., 2024)	self-supervised	384 / 768 / 1024 / 1536	5 each
AugReg ViT tiny / small / base / large (Steiner et al., 2022)	class labels (21k then 1k)	192 / 384 / 768 / 1024	5 each
ViT base / large (Dosovitskiy et al., 2021)	class labels (1k)	768 / 1024	5 each
OpenCLIP B/32, B/16, L/14, H/14, g/14 on LAION-2B (Cherti et al., 2023)	image-text contrastive	768 / 768 / 1024 / 1280 / 1408	5 each
CLIP B/32, B/16, L/14 (Radford et al., 2021), SigLIP (Zhai et al., 2023), SigLIP 2 (Tschannen et al., 2025)	image-text contrastive	768 / 768 / 1024 / 768 / 768	5 each
Pythia 70M to 6.9B (seven sizes) (Biderman et al., 2023)	next token	512 to 4096	7 each (5 for subsumption)
Qwen2.5 0.5B / 1.5B / 3B / 7B (Yang et al., 2024), Qwen 1.5B / 7B (Bai et al., 2023)	next token	896 / 1536 / 2048 / 3584, 1536 / 3584	7 each (5 for subsumption), final
GPT-2 small / medium / large / XL (Radford et al., 2019)	next token	768 / 1024 / 1280 / 1600	7 each (5 for subsumption)
Llama-3.1-8B (Grattafiori et al., 2024), Gemma-2-2B (Gemma Team, 2024), Gemma-3-1B / 4B / 12B (Gemma Team, 2025)	next token	4096 / 2304 / 1152 / 2560 / 3840	7 each (5 for subsumption)

Table 3: **The networks.** Layers are sampled at fixed relative depths from input to output, and "final" denotes the last stored layer only. The size ladders were added for the effect-of-size study. The toy trained on independent random labels is the no-hierarchy control.

B THE OTHER PICTURES OF HIERARCHY

The main text follows the nested subspace hypothesis. Several literatures assume other pictures of how a network could lay out concept hierarchy. Each is a *model* in the scientific sense, a committed claim about geometry, and each implies a *signature*, a statistic that must be high if the model holds. Table 4 lists them with their origin and verdict under the controls of Section 2. The remaining signatures are rank correlations (norm against tree depth, prefix energy against tree depth, tree distance against embedding distance) or box-containment rates.

The tree that is nested. An independent biological taxonomy over the same images, the NCBI classification matched by Latin binomials (103 species, 310 taxa), tests whether the nesting follows any tree or the semantic one. Against the shuffled pairing, NCBI reads at its null beyond its finest rank in four vision encoders (DINOv2-B, ViT-B, CLIP, SigLIP), from $+0.02$ to $+0.06$ at the family rank and from -0.02 to $+0.01$ at the order, class and phylum ranks, while WordNet over the same species and with the same estimation quality (every held-out parent with at least 150 samples) reads $+0.12$ to $+0.15$.

Signatures are specific. On synthetic activations in which hierarchy is written in six different ways (12 seeds), subspace containment responds only to nested subspaces, box containment only to boxes, and prefix energy only to prefixes. Subsumption firing and tree distance respond to any hierarchy, and the no-hierarchy control reads near zero for every signature (Figure 6, left). Encoders trained under each literature's actual objective reproduce the pattern, and the profile of real networks matches ancestor-closure supervision and no other objective (Figure 6, right).

On real activations (seven models, three seeds each) the prefix and box signatures read ≈ 0 in every model, strict cone nesting sits at its random baseline (see the strict-cones paragraph below), norm-as-depth is present (0.19 to 0.53), and subsumption firing is the largest signal everywhere. Rebuilding the WordNet tree under each picture's own rule makes the verdicts concrete. Prefix and box rules recover almost nothing (ancestor-F_1 0.05 to 0.26 against 0.58 to 0.88 for containment). The norm rule is the runner-up, but given its real machinery (a Poincaré-ball embedding of the same

Model	Origin	Signature (what must be high)	Verdict
M1 No structure	null model	nothing, every signature at its null	refuted
M2 Depth as norm	hyperbolic / Poincaré embeddings (Nickel & Kiela, 2017)	depth in the tree correlates with activation norm	partial, heuristic
M3 Nested cones	entailment cones (Ganea et al., 2018)	child's cone inside parent's cone	baseline in real models
M4 Box containment	order / box embeddings (Vendrov et al., 2016; Vilnis et al., 2018; Li et al., 2019)	child's axis-aligned box inside parent's box	**absent**
M5 Coordinate prefixes	Matryoshka representations (Kusupati et al., 2022)	discriminative variance moves to later coordinates with depth	**absent**
M6 Ultrametric distance	hierarchical clustering and taxonomy readouts (Kusupati et al., 2021)	tree distance recoverable from pairwise distances	partial (generic)
M7 **Nested subspaces**	this work, and the subspace analyses of Park et al. (2025)	child's PCA subspace inside parent's PCA subspace	**alive everywhere**
M8 **Subsumption firing**	Sanderson & Croft (1999), distributional inclusion (Weeds & Weir, 2003; Geffet & Dagan, 2005; Kotlerman et al., 2010), box probabilities (Vilnis et al., 2018), tree-structured sparsity (Jenatton et al., 2010)	P(parent fires | child fires) high, converse low	**alive everywhere**

Table 4: **Pictures of hierarchy proposed in the literature**, the signature each implies, and its verdict under the controls of Section 2. M8 is probabilistic rather than geometric, the subsumption definition of Sanderson & Croft (1999) and of the Distributional Inclusion Hypothesis. M4 and M5 are stated in the network's own coordinate basis.

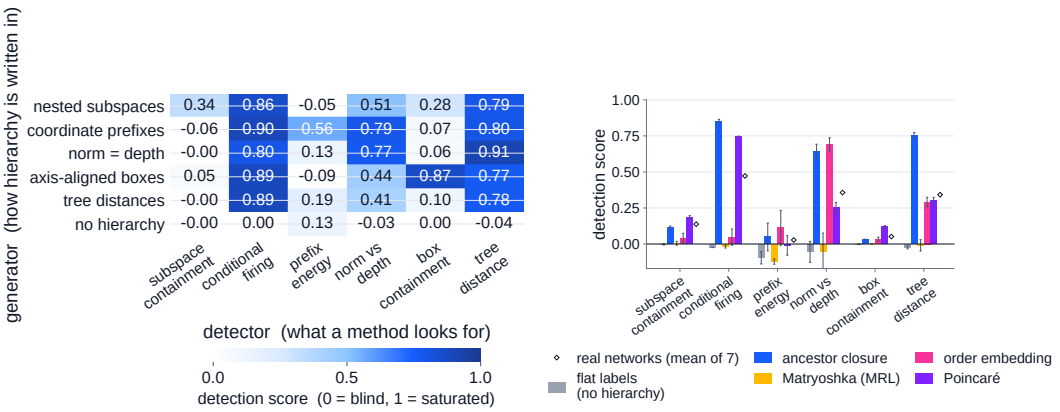

Figure 6: **Which signature sees which picture.** (Left) Each signature measured on synthetic activations in which hierarchy is written in as each of the six pictures. (Right) The same signatures on toy encoders trained under each literature objective, with diamonds marking the average of the real networks.

similarities decoded by its own norm and cone rules) it collapses to an ancestor-F_1 of at most 0.05 (Appendix A), so what survives of M2 is the raw norm as a depth heuristic, not hyperbolic geometry.

A parent probe on unseen subclasses. We take the 30 parents with the most children, among parents with at least three children of at least 30 samples each. A third of each parent's children is held out (at least one). A multinomial logistic regression ($C = 1$) on centred, unit-norm activations learns to name the parent from up to 400 samples of its other children. It is tested on up to 150 samples of the held-out children, which it has never seen. It is also tested on 150 unused samples of the children it was trained on, over three random splits. Chance is $1/30 \approx 0.033$, and the largest class holds 0.034 of the test samples in vision and 0.039 to 0.041 in GPT-2. At the output, accuracy on unseen children is 0.34 in DINOv2-B, 0.35 in ViT-B, 0.37 in OpenCLIP B/32 and 0.39 in SigLIP and in CLIP B/16. This is ten to twelve times chance, against 0.58 to 0.67 on the children it was trained on. In GPT-2 it is 0.15 from the third layer to the last, four and a half times chance, against 0.42 on trained children. In the vision encoders it rises from chance at the first stored layer to its highest values in the last layers. What a representation knows about *dog* is available for a breed it has never labelled.

Strict cones. The strict cone-nesting signature (M3, every child activation inside the parent cone) sits at its random baseline in all real models tested, while being satisfiable in toy models, consistent with cones being a special case that real networks do not use, even where subspace nesting is strong.

C Building the Geometry into a Featurizer

If nested subspaces are the right picture, a featurizer that carries that prior should express the tree at little cost. Sparse autoencoders are the natural test bed, since hierarchy is already their known failure mode in the form of feature splitting (Bricken et al., 2023) and absorption (Chanin et al., 2024). We compare sparse-autoencoder architectures at matched dictionary size and sparsity on toy, Qwen-1.5B, DINOv2 and CLIP activations, three seeds each. The architectures are vanilla TopK, an L_1 SAE, Matryoshka (Bussmann et al., 2025), a tree-gated SAE that hard-wires subsumption, an additive subsumption penalty, a coarse-to-fine "ladder" of latent levels in which each level encodes the residual left by the coarser ones, and a supervised oracle whose code is forced to carry the true ancestor chain. Each dictionary is scored on tree recovery from its latents alone (the induction rule of Section 4), on absorption (how often the parent's latents are silent on a child sample that the child's latents detect) and on reconstruction cost (Table 5).

Findings. Nested scales recover the tree. The ladder reaches an ancestor-F_1 of 0.37 to 0.40 on all four datasets against 0.07 to 0.34 for vanilla TopK, at 0.60 to 0.94 of its error and at the level of the forced-support oracle (0.31 to 0.38), which pays only 4 to 9% additional error for carrying the ancestor chain. Enforcing the subsumption order does not help recovery. Parent gates (Jenatton et al., 2010) and an additive parent-child penalty bind the latent order to the tree up to sixteen times above a shuffled null, while recovery stays where vanilla TopK leaves it. Absorption is not removed, since the ladder absorbs as often as vanilla TopK (0.32 to 0.57 against 0.30 to 0.56, and the oracle 0.01 to 0.16). Ordinary dictionaries already inherit the geometry. On Qwen the parent's latent fires on the child's samples at 0.60 to 0.64, far above a matched null, and Gemma Scope's public dictionary on layer 12 of Gemma-2-2B decodes the tree at an ancestor-F_1 of 0.33, the linear-probe ceiling for that layer.

D The Effect of Depth and Size

Both measurements across depth and size. Figure 7 gives both measurements across depth in the vision encoders, the full version of the left panel of Figure 5, and Table 6 the fits against size at the output, at a matched relative depth and at each vision encoder's peak layer.

Section 4 reads the size ladders at the output layer. This appendix gives the same fits at a matched relative depth and at each encoder's peak layer, the paired comparison of the largest and the smallest model of every family, the tree rebuilt from activations at every size, and the sensitivity of the smallest widths to the dimension cap. Every model of a modality scores the same edges, so the paired differences are far tighter than the independent intervals. Where a family's medians approach zero, the fitted exponent is not meaningful and the paired difference is the reading.

Matched depth and peak. At relative depth 0.75 the decline of tree-specific containment with size disappears in DINOv2 and in both CLIP families and remains in the AugReg ladder, and at the peak

	Toy			Qwen-1.5B			DINOv2			CLIP		
architecture	recov.	absorp.	cost	recov.	absorp.	cost	recov.	absorp.	cost	recov.	absorp.	cost
vanilla TopK (L_0 32)	0.10±0.00	0.37	1.00	0.34±0.01	0.53	1.00	0.07±0.01	0.34	1.00	0.14±0.01	0.31	1.00
vanilla TopK (L_0 54)	0.10±0.02	0.34	0.74	0.35±0.01	0.56	0.92	0.08±0.00	0.37	0.89	0.11±0.01	0.30	0.83
L_1 SAE	0.15±0.02	0.13	0.66	0.30±0.03	0.32	0.76	n/a	n/a	n/a	n/a	n/a	n/a
Matryoshka	0.24±0.09	0.16	0.96	0.40±0.04	0.51	1.08	0.21±0.06	0.45	1.14	0.39±0.01	0.25	1.19
subsumption penalty	0.13±0.01	0.30	1.05	0.28±0.09	0.60	1.01	0.08±0.01	0.34	1.04	0.12±0.01	0.31	1.04
tree-gated	0.33	0.25	3.17	0.29	0.29	3.11	0.28	0.25	3.38	0.30	0.18	3.33
ladder (L_0 32)	0.24±0.00	0.39	0.93	0.36±0.04	0.54	1.02	0.18±0.07	0.37	1.06	0.35±0.02	0.35	1.05
ladder (L_0 54)	0.39±0.01	0.42	0.60	0.37±0.00	0.57	0.94	0.37±0.01	0.42	0.94	0.40±0.02	0.32	0.86
forced-support oracle	0.37±0.01	0.01	1.05	0.38±0.02	0.16	1.04	0.34±0.00	0.14	1.06	0.31±0.02	0.15	1.09
exact-indicator features	0.38			0.18			0.30			0.30		
linear-probe features	0.42			0.36			0.38			0.36		

Table 5: **Sparse-autoencoder architectures at matched dictionary size.** For the toy transformer, Qwen-1.5B, DINOv2 and CLIP activations, tree recovery from the latents alone (ancestor F_1, mean and standard deviation over three seeds where available), the absorption rate (the fraction of child samples detected by the child's latents on which the parent's latents are silent) and reconstruction cost (mean squared error relative to vanilla TopK at the same sparsity). The two reference rows give recovery from exact-indicator and from linear-probe features. Higher recovery, lower absorption and lower cost are better.

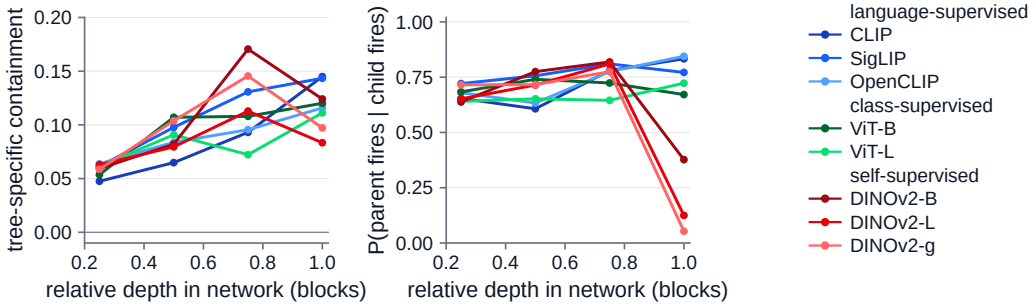

Figure 7: **Both measurements across depth in the vision encoders.** One line per encoder, coloured by training target (blues language-supervised, greens class-supervised, reds self-supervised). Depth counts transformer blocks from the first block to the output, and the earliest blocks are not drawn. The input embedding is left out, since its class token is the same vector for every image. (Left) Tree-specific containment, the held-out value minus its value under a shuffled pairing. (Right) Held-out subsumption, how often the parent's feature fires when the child's does.

layer no family declines (Table 6). The self-supervised readout, which falls by an order of magnitude with size at the output, is flat at relative depth 0.75.

Paired differences. The largest model of each vision ladder is a few hundredths less tree-specific than the smallest on the same edges, the language model ladders show no paired difference beyond two hundredths, and only the DINOv2 ladder shows a sizeable paired loss of the readout (Table 6).

Recovery and the cap. The tree rebuilt from activations under the containment rule is as good or better in the largest model of a ladder as in the smallest (ancestor-F_1 0.70, 0.78, 0.83 and 0.81 from DINOv2 small to giant, and 0.70 to 0.78 across the OpenCLIP ladder). Repeating the analysis with the PCA cap lowered from 80 to 40 dimensions, which matters for the widths of 192 and 384, changes no exponent by more than 0.03.

E Curvature inside the Spans

Tests. Every measurement in the main text is linear, so it shows that hierarchy is carried by nested linear spans and cannot say whether the clouds are also curved inside those spans. Four tests, in the robustness log with a synthetic sanity check on six generators (nested flat subspaces, a nested torus, nested spherical caps, nested balls, span-only nesting, and no hierarchy), address this on the stored activations of the twenty-three networks in the battery before the size ladders, at every stored layer, with the same edges, held-out parents and shuffled pairings as the anisotropy audit. First, the intrinsic

		tree-specific containment		held-out subsumption	
family	sizes	exponent b	largest minus smallest	exponent b	largest minus smallest
output layer					
DINOv2 (self-supervised)	4	−0.16 [-0.22, -0.11]	−0.037 [-0.050, -0.023]	−0.57 [-0.93, -0.42]	−0.210 [-0.292, -0.163]
AugReg ViT (class-supervised)	4	−0.16 [-0.20, -0.10]	−0.041 [-0.060, -0.022]	+0.00 [-0.03, +0.03]	+0.000 [-0.018, +0.014]
ViT (class-supervised)	2	(2 sizes)	−0.007 [-0.013, -0.000]	(2 sizes)	+0.000 [+0.000, +0.007]
OpenCLIP LAION-2B (language-supervised)	5	−0.10 [-0.12, -0.05]	−0.026 [-0.033, -0.017]	−0.01 [-0.03, -0.00]	−0.010 [-0.022, -0.002]
OpenAI CLIP (language-supervised)	3	−0.14 [-0.23, -0.06]	−0.024 [-0.037, -0.017]	−0.02 [-0.04, +0.01]	−0.004 [-0.013, +0.000]
Pythia (next token)	7	+0.09 [-0.04, +0.71]	+0.016 [-0.006, +0.033]	+0.12 [+0.04, +0.19]	+0.051 [+0.018, +0.089]
Qwen2.5 (next token)	4	−0.06 [-0.34, +1.05]	−0.010 [-0.024, +0.004]	+0.08 [+0.04, +0.14]	+0.025 [+0.014, +0.042]
GPT-2 (next token)	3	+0.16 [-0.06, +0.55]	−0.007 [-0.020, +0.012]	(2 sizes)	+0.000 [-0.002, +0.000]
Gemma 3 (next token)	3	−0.16 [-0.24, -0.01]	−0.015 [-0.025, -0.006]	+0.01 [-0.10, +0.06]	−0.001 [-0.011, +0.000]
relative depth 0.75					
DINOv2 (self-supervised)	4	−0.03 [-0.08, +0.02]	−0.006 [-0.029, +0.007]	−0.02 [-0.03, -0.01]	−0.025 [-0.048, -0.007]
AugReg ViT (class-supervised)	4	−0.19 [-0.24, -0.13]	−0.070 [-0.086, -0.041]	−0.04 [-0.06, -0.03]	−0.074 [-0.090, -0.056]
ViT (class-supervised)	2	(2 sizes)	−0.033 [-0.044, -0.024]	(2 sizes)	−0.040 [-0.066, -0.021]
OpenCLIP LAION-2B (language-supervised)	5	−0.07 [-0.14, +0.05]	+0.006 [-0.011, +0.016]	+0.01 [-0.03, +0.03]	−0.004 [-0.032, +0.008]
OpenAI CLIP (language-supervised)	3	+0.01 [-0.12, +0.21]	+0.021 [+0.006, +0.035]	−0.03 [-0.05, +0.01]	+0.001 [-0.015, +0.008]
Pythia (next token)	7	+0.04 [-0.10, +0.45]	−0.018 [-0.034, +0.021]	+0.06 [-0.04, +0.16]	+0.000 [-0.000, +0.001]
Qwen2.5 (next token)	3	+0.00 [+0.00, +0.77]	+0.000 [-0.007, +0.007]	n/a	n/a
GPT-2 (next token)	4	+1.37 [+0.12, +1.73]	+0.001 [-0.017, +0.026]	+0.28 [+0.00, +0.56]	+0.000 [+0.000, +0.004]
Gemma 3 (next token)	3	−0.10 [-0.24, +0.06]	−0.006 [-0.026, +0.005]	+0.10 [+0.00, +0.18]	+0.005 [+0.000, +0.017]
peak layer (vision only)					
DINOv2 (self-supervised)	4	−0.00 [-0.09, +0.07]	+0.000 [-0.011, +0.013]	+0.02 [+0.01, +0.04]	+0.040 [+0.022, +0.057]
AugReg ViT (class-supervised)	4	+0.01 [-0.09, +0.12]	−0.001 [-0.016, +0.022]	−0.01 [-0.01, +0.00]	+0.001 [-0.011, +0.014]
ViT (class-supervised)	2	(2 sizes)	−0.006 [-0.020, +0.011]	(2 sizes)	−0.029 [-0.041, -0.015]
OpenCLIP LAION-2B (language-supervised)	5	+0.06 [-0.13, +0.19]	−0.013 [-0.032, -0.000]	−0.05 [-0.09, -0.02]	−0.048 [-0.072, -0.012]
OpenAI CLIP (language-supervised)	3	+0.23 [+0.01, +0.48]	+0.007 [-0.014, +0.028]	−0.04 [-0.08, -0.01]	−0.008 [-0.021, +0.009]

Table 6: **The effect of size, family by family.** Fitted exponent b of a power law in the parameter count, with its cluster-bootstrap interval, and the paired difference between the largest and the smallest model of the family on the same edges. Families with two sizes show the paired difference only. Models whose final layer is degenerate or whose rate-matched null saturates are left out of the subsumption fit. Where a family's medians approach zero the exponent is not meaningful and the paired difference is the reading.

dimension of each concept cloud (TwoNN and a maximum-likelihood estimator, computed inside the cloud's own span) is compared with that of a Gaussian cloud of the same covariance, since a curved sheet has a lower intrinsic than linear dimension. Second, region containment (the fraction of child samples inside a nearest-neighbour support of the held-out parent, with its threshold set on the parent's own samples) is compared with span containment through the paired probability that the true parent scores above a shuffled parent. Third, chords between far-apart samples of a cloud are scored by how far their interior points fall from the data, relative to the same statistic on the Gaussian match, since chords of a curved surface leave it while chords of a blob do not. Fourth, local tangent spaces at child samples are tested for containment in the parent's local tangent spaces at the same points.

Curved clouds, linear hierarchy. The clouds are curved, and the hierarchy is not. Inside its own span a concept cloud has an intrinsic dimension well below that of a Gaussian cloud with the same covariance, 0.60 of it in DINOv2, 0.71 in ViT, 0.65 in CLIP and 0.42 in Qwen-1.5B at the output, with every interval excluding 1, which amounts to about nine to eleven intrinsic dimensions inside spans of about thirty in the vision encoders. Chords between far-apart samples leave the data more than chords of the Gaussian match do, by factors of 1.40, 1.41, 1.21 and 1.25 in the same four networks, and parent clouds behave the same way. The curvature grows with depth, from a dimension ratio of 0.89 and a chord factor of 1.10 in the second layer of DINOv2 to 0.60 and 1.39 at its output. None of this carries hierarchy beyond the spans. Region containment, which asks whether the child's samples fall inside the parent's own support, separates true parents from shuffled parents worse than span containment does in every network (a paired probability of 0.38 to 0.65 for the region against 0.85 to 0.94 for the span in the vision encoders, and 0.53 against 0.63 in Qwen-1.5B), and its tree-specific excess is zero in DINOv2 and ViT (0.28 in CLIP, against a span excess of 0.14 that discriminates better). Local tangent nesting equals global span nesting to within 0.08 in every network. The tests detect curvature on a nested torus and on nested spherical caps, detect region nesting beyond spans on nested balls, and read flat on nested flat subspaces.

The full grid of networks says the same. Of the twenty networks with a usable final layer, eighteen have an intrinsic dimension below their Gaussian match with the interval excluding one, and seventeen fall below 0.7 of it, the language models lowest at 0.35 to 0.48. Chords leave the data more than in the Gaussian match in all twenty, and by more than 1.3 in the six class-supervised and self-supervised

vision encoders (up to 2.0 in DINOv2 giant), against 1.17 to 1.21 in the language-supervised encoders and 1.13 to 1.29 in the language models. No network shows region containment separating parents better than span containment, and the gap is clearly negative in eighteen. Region containment is alive on its own only in the language-supervised encoders (excess 0.28 to 0.34) and weakly in two toys, where the span still discriminates better. Local tangents nest as the global spans do in every network. Across depth the curvature grows toward the output in the vision encoders and is flat in Llama-3.1-8B, while the region test ties with the span test in early and middle layers and loses to it at the output. The anisotropic layers in which every concept contains every other (the middle layers of Qwen2.5-7B and the final layer of GPT-2-medium) are excluded, since no containment statistic is interpretable there.

The nested subspace hypothesis is therefore the right level of description for the hierarchy. The clouds inside the spans are curved, so a probe or a featurizer that assumes Gaussian blobs will misjudge their shape, but a probe that reads the span reads the hierarchy.