ATTRIBUTING CLASS RECOGNITION TO INVARIANTS WITH RECURSIVE PEELING

Anonymous authors
Paper under double-blind review

ABSTRACT

A sparse autoencoder splits what a vision model computes into concepts, and most work studies these concepts one at a time. We ask instead what they reveal together. We call them class unique when they fire consistently on one class and shared when they fire consistently on several, representing invariants of different sizes. We propose Multi-level Clustering with Peeling (MCP) to uncover how these invariants organize classes and contribute to recognition. MCP clusters classes at resolutions determined by how widely their concepts are shared. Concepts shared only within one cluster are peeled as cluster unique concepts, exposing further relationships at subsequent levels. Applied to the final-layer class token of the self-supervised DINOv2 on ImageNet, MCP reveals a forest of visual categories that departs from the WordNet hierarchy. The clusters are tighter and more semantically coherent than size-matched random clusters. Ablations with a frozen linear classifier show that removing the concepts peeled from a cluster lowers accuracy within that cluster, with near-zero median effects outside it. Different classes rely on class unique invariants, cluster unique invariants, or both. Removing both kinds produces super-additive accuracy losses, while either kind alone can preserve recognition. These results suggest that the model recognizes classes by parallel abduction, weighing evidence for single classes and for clusters of related classes at once. Class unique invariants distinguish individual classes, while cluster unique invariants support several competing classes. Unlike human vision, which reaches its decision by recursive abduction, the model weighs every candidate in a single pass.

1 INTRODUCTION

Sparse autoencoders decompose what a vision model computes into latent features that people can inspect. The autoencoder's weights form dictionary atoms, and a latent feature fires when its activation is nonzero. We call the combination of a dictionary atom and a latent feature a concept, which identifies a learned invariant. Most existing work studies these concepts one at a time, naming a latent feature and testing its effect on a class (Bau et al., 2017; Kim et al., 2018). Two questions stay open when concepts are read one at a time. The first is what structure the concepts form among classes. A concept that fires consistently on several classes ties those classes together. When many concepts do so, the class structure they induce is itself a discovery about the model. Humans organize the ImageNet labels into a single taxonomy that follows the WordNet hierarchy. We ask whether the ViT model organizes its classes the same way or into something looser. Human vision recognizes an object by abduction, inferring the explanation that best accounts for what it sees. It reaches this inference recursively. A first glance gives a coarse category and a set of likely interpretations, and a closer look then settles the details (Hochstein & Ahissar, 2002; Bar, 2003). A classic model of the first glance learns a dictionary of shape features shared across objects, with category-specific circuits at the top (Serre, 2006). Our second question is how a vision transformer, which sees an image in a single pass, reaches its decision from invariants shared by one class or by many. Both answers matter to anyone who audits, edits or compares a vision model through its concepts. Whether a class has concepts of its own decides whether a class-specific explanation of it exists at all.

Answering the first question requires reading class structure at more than one scale. A concept shared by 2 classes and a concept shared by 36 describe structure at different resolutions, and no single clustering captures both. Prior work supplies the hierarchy as an input (Olson et al., 2025; Nguyen et al., 2026), so whether concept structure forms one has not been

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.

tested as an output. Answering the second question requires measuring the causal effect of a class's invariants through ablation, class by class rather than as one population.

We focus on vision transformers to study recognition in an attention-based architecture also used by large language models. We use DINOv2 (Oquab et al., 2024), a self-supervised vision transformer that learns its features without class labels. We study it on ImageNet-1k (Russakovsky et al., 2015), which contains 1.28 million training images across 1,000 classes. Only the linear classifier we train on its class token sees the labels, so the invariants we find are the model's own. This lets us measure which concepts recur within each class and which are shared across classes. We analyze the class token after the model's final layer, where it summarizes the whole image. We call a concept class unique when it fires consistently on exactly one class, and shared when it fires consistently on two or more. Classes that share several concepts form a cluster, and the shared concepts whose remaining classes all lie in that cluster are called cluster unique concepts. A class unique concept is an invariant of one class. A cluster unique concept is an invariant of the classes it fires consistently on. To identify invariants with different sizes at different resolutions, we propose Multi-level Clustering with Peeling (MCP). MCP clusters classes from fine to coarse levels and peels cluster unique concepts between levels. It assigns every shared concept to one cluster at one level, and every class it fires consistently on lies in that cluster. That assignment is what lets a cluster's concepts be removed together and the effect be read on the classes they fire on.

We make four contributions.

- We propose MCP, a multi-level clustering method that identifies cluster unique concepts level by level at resolutions read from the data. The method does not limit the number of children each node may have, nor does it dictate how many classes each cluster may have.
- We show that the clusters form a forest structure rather than one taxonomy. At every level where each test applies they are tighter and more WordNet-coherent than size-matched random clusters.
- We show that invariants of different sizes identified by MCP are causal for classification, and that their effect differs by cluster size. These results suggest that the model recognizes classes by parallel abduction, weighing evidence for single classes and for clusters of related classes at once, in contrast with the recursive abduction of human vision.
- We take a statistical approach throughout, testing every claim against an empirical null distribution or a matched control rather than against selected cases.

Related work. Concept-based analysis of vision models is well developed at the level of single concepts. Early work aligns latent features to a vocabulary of visual patterns (Bau et al., 2017) and tests whether a named concept influences a class prediction (Kim et al., 2018). Later work discovers concepts from the data by factorizing activations (Fel et al., 2023) or by training sparse autoencoders on vision transformers (Fel et al., 2025; Stevens et al., 2025; Joseph et al., 2025; Pach et al., 2025). All of this work treats each concept in isolation.

Structure across concepts is far less developed. One line organizes concepts by network depth (Kowal et al., 2024). Another starts from a hierarchy. The closest work tests whether DINOv2 class-token features align with the ImageNet taxonomy (Olson et al., 2025). Hierarchical concept pursuit takes a hierarchy of semantic concepts as given and recovers concepts along its paths (Nguyen et al., 2026). Other methods build the hierarchy into the autoencoder itself. Matryoshka autoencoders train nested dictionaries of increasing size, cascaded autoencoders train a second autoencoder on the first one's decoder weights, and tree autoencoders learn a tree over the latent features (Bussmann et al., 2025; Zhao et al., 2026; Cao et al., 2026). In each case the hierarchy is supplied in advance or fixed by the architecture, so nesting is assumed rather than discovered. We organize concepts instead by the classes they fire consistently on. A concept that fires consistently on a class stays fixed while the class's images vary. That is the sense of invariant used in causal and representation learning (Peters et al., 2016; Schölkopf et al., 2021). A shared concept is an invariant of several classes at once, and those concepts give the structure we read out.

The latent features an autoencoder finds also depend on the training run (Paulo & Belrose, 2025). Unstable features carry little of the signal and the stable ones carry most of it (Gerasimov et al., 2026). Each architecture also favors some kinds of concept over others (Hindupur et al., 2025), so structure read from one dictionary has to be shown again on another. Validation in this area still rests largely

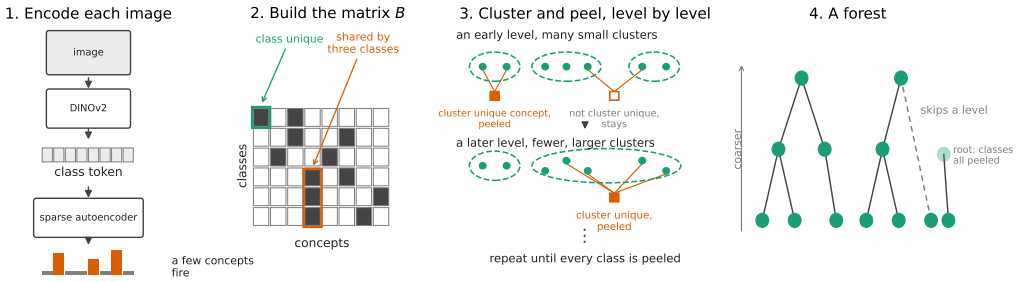


Figure 1: The pipeline. (1) A sparse autoencoder encodes image class tokens into concepts. (2) Class-level firing rates define the biadjacency matrix B. (3) MCP alternates clustering and peeling. (4) Containment links clusters across levels into a forest. Dashed edges mark parents found beyond the next level. Lighter nodes mark roots.

on inspecting selected examples (e.g., Ghorbani et al., 2019; Joseph et al., 2025; Pach et al., 2025). Tests against a null distribution or a matched control are rare.

No prior work asks whether shared concepts organize a model's classes into multi-level clusters on their own. None asks whether the invariants so found are causal for classification, or which kind of invariant each class is recognized through. We close these gaps.

2 Multi-level Clustering with Peeling

Figure 1 shows the pipeline for identifying how concepts connect classes. First, a sparse autoencoder encodes each image's final-layer class token into a few concepts. The fraction of a class's images on which each concept fires determines the links in the biadjacency matrix B (Section 2.1). MCP uses these links to cluster classes from fine to coarse levels, peeling cluster unique concepts between levels (Sections 2.2 and 2.3). Finally, clusters are linked across levels by containment of their remaining classes to form a forest (Section 3.1).

2.1 Concepts and the biadjacency matrix B

We train a sparse autoencoder with the archetypal architecture (Fel et al., 2025) on DINOv2's class token over the 1.28M ImageNet training images. We choose the archetypal architecture because it constrains each dictionary element to a convex combination of data points. The autoencoder learns 8,192 concepts and keeps the eight strongest per image through top-k selection (Gao et al., 2025). Eight concepts reconstruct the token at a normalized mean squared error of 0.24 on validation images, so most of what the token carries survives in eight concepts per image. The remaining constants are in Appendix A. Autoencoder features can change between training runs (Paulo & Belrose, 2025), so we repeat the peel and the causal tests on a second autoencoder trained from an independent initialization (Appendix A).

A link between a class and a concept must be consistent across the class's images rather than an accident of one image. We therefore decide links by a frequency over the whole class. The firing rate of concept j on class c is the fraction of the class's training images on which the concept fires. Class c is linked to concept j when this rate reaches 0.2. Formally,

$$r_{c,j} = \frac{1}{n_c}\left|\{\, i \in I_c : z_{i,j} > 0 \,\}\right|, \qquad B_{c,j} = \mathbf{1}\left[\, r_{c,j} \geq 0.2 \,\right], \tag{1}$$

where I_c holds the n_c training images of class c and $z_{i,j}$ is the activation of concept j on image i after the selection of the eight strongest. The links form the biadjacency matrix B, with one row per ImageNet class and one column per concept. Entry $B_{c,j} = 1$ records the link.

The choice of 0.2 as the rate threshold is motivated by two findings. First, the concepts below it carry little function. Removing all 2,185 of them costs 2.32 pp of top-1 accuracy, while random sets with the same total activation cost 8.08 pp. Second, every higher threshold we tested leaves some class

with no link at all. The value 0.2 is thus the largest tested threshold that keeps every class linked (Appendix D).

B fixes the terms used in the rest of the paper. A link makes concept j an invariant of class c, since the concept fires on at least a fifth of the class's images. A concept linked to exactly one class is class unique, and a concept linked to two or more classes is shared. The number of classes a concept links is its firing set size (FSS), representing the size of the invariant. The matrix holds 3,052 class unique and 1,227 shared concepts, and only the latter can tie classes together. Matrices rebuilt from different image samples agree at the level a counting model of sampling noise predicts, so B is as stable as its sample size allows (Appendix D).

2.2 Resolution levels from FSS

The nontrivial choice is the sequence of clustering resolutions used while the matrix changes. The resolution of a level is its number of clusters K. A large K produces many small clusters and exposes invariants shared by few classes, while a small K produces few large clusters, which broadly shared invariants need. A preset grid of K values has one issue. It loses the intended peeling order from narrow to broad invariants as classes leave B. Write a_l for the number of classes still active at level l, where a class is active while at least one of its invariants is unpeeled. K clusters then hold a_l/K classes each on average, so the same K becomes finer as a_l decreases.

Two counts must be kept apart, and one equation links them to the levels. For concept j and level l,

$$\mathrm{FSS}_j = \sum_c B_{c,j}, \qquad f(s) = \big|\{\, j : \mathrm{FSS}_j = s \,\}\big|,$$
$$s_l = \text{the } l\text{-th smallest } s \geq 2 \text{ with } f(s) > 0, \qquad K_l = \max\big\{1,\ \lfloor a_l/s_l \rceil\big\}. \tag{2}$$

FSS_j counts the classes that concept j links to, and not the images it fires on. The frequency $f(s)$ counts the number of concepts with FSS equal to s, from $f(2) = 786$ to $f(36) = 1$ (Figure 2a). The levels visit each nonzero $f(s)$ once, read from the full matrix B, so s_l is the number of classes that an invariant targeted at level l shares. We choose K_l so that a cluster contains s_l classes on average. Therefore, K_l is mainly decided by the distribution of FSS. Level 0 removes the class unique concepts without clustering (Section 2.3), and the clustering levels are numbered from 1. The ratio a_l/s_l is rarely an integer, so $\lfloor\cdot\rceil$ rounds it to the nearest integer, ties to even.

Once B is fixed, the schedule has no tuned parameter, since the anchors s_l are read from the matrix's own FSS values. Of four schedules compared on B, only a preset grid that ignores the remaining classes loses the broadest invariants (Appendix E).

2.3 The peeling loop

Restricted to the shared concepts, B has 929 rows and 1,227 columns, since 929 of the 1,000 classes carry a shared concept. The clustering step therefore has fewer points than dimensions. Each class row is sparse as well, since the median class carries 3 of the 1,227 concepts and 99.3% of class pairs share no concept at all. In this regime a distance between two rows reflects mostly what the pair does not share, so distance-based clustering is unreliable. We therefore use sparse subspace clustering (SSC), which is designed for sparse high-dimensional data (Elhamifar & Vidal, 2013). It expresses each class's row as a sparse weighted sum of the other classes' rows. A class can be rebuilt mainly from classes that carry the same invariants, so the nonzero weights point to the classes it shares invariants with. We compute the weights with a Lasso regression, and a spectral step clusters the resulting adjacency between classes (Damle et al., 2019). We use the same Lasso penalty at every level and in every build.

There are three motivations for the peeling step. First, peeling lets each level look for invariants shared by a different number of classes. A peeled concept is pinned to the cluster it leaves as a cluster unique concept of that cluster. On the matrix B, 1,197 of the 1,227 shared concepts are peeled once the target cluster size reaches their FSS. Second, small invariants act as noise when clustering at larger target sizes, and peeling removes those that finer levels have already explained. Without peeling, every level clusters the same adjacency, which is dominated by the 786 concepts with FSS 2. Coarser levels then mostly merge finer clusters. Only 21% of clusters split at the next level, against 57.5% with peeling. Third, peeling reduces the number of concept dimensions. The ratio of remaining

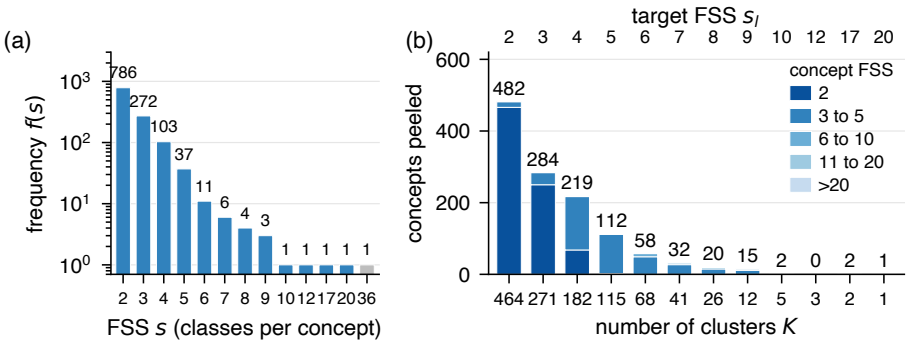


Figure 2: (a) Frequency $f(s)$, the number of concepts with FSS s, over the 1,227 shared concepts. Gray marks FSS 36, which sets no level since K_l is already 1 at $s_l = 20$. (b) Concepts peeled per level, stacked by concept FSS, with K_l below and s_l above. Bar labels are peeled counts and sum to 1,227.

classes to remaining shared concepts rises from 0.8 at the first level to 5.6 at level 8, limiting the impact of the curse of dimensionality. Appendix B.2 gives the evidence for each reason.

We first peel all class unique concepts because they provide no links between classes. At each subsequent level, SSC clusters the remaining classes at the resolution defined in Section 2.2. A cluster unique concept is a concept whose remaining linked classes all fall within one cluster. These concepts are peeled before moving to the next level. If all concepts of a class have been peeled, the class is peeled too. The process repeats until all classes are peeled. On B, MCP peels all 1,227 shared concepts over levels from $K_1 = 464$ to $K_{12} = 1$, narrow invariants first (Figure 2, Appendix B). Only one concept is left for the last level, whose single cluster peels whatever remains by construction.

3 Structure of the Clusters

Across levels MCP's clusters form a forest of trees rather than one taxonomy. They are also tighter than random clusters of the same sizes. The structure is therefore not an artifact of the cluster sizes MCP sets.

3.1 Peeling Coverage and the Forest

The clustering at each level does not require a coarse cluster to contain a fine one. After peeling ends, a cluster's parent is the first coarser cluster containing all of its classes that are still active. A parent at the next level is called an immediate parent. The parent search stops once fewer than two of the cluster's classes remain active. Linking each cluster to its parent gives the forest shown in Figure 1, panel 4.

Most clusters have a parent at a coarser level. Of the 1,039 clusters with at least two active classes before the last level, 69.2% find a parent containing all their still-active classes. An immediate parent is found for 36.5%, while 32.7% find one at a later level. The remaining 30.8% have no parent. Their classes are peeled before any coarser cluster contains them together, so no coarser cluster holds them as one group. Each root and the clusters connected to it form a tree. The 71 of 1,000 classes with no shared concept form single-node trees. Together, these trees form the forest, which is different from the WordNet hierarchy. The forest has no dominant tree. The largest holds 122 of the 929 classes with a shared concept, and 11 trees hold 20 or more. The forest's 321 roots start separate trees, so the forest holds many categories at once rather than one path from coarse to fine. The procedure alone splits fewer clusters. On a synthetic matrix holding one nested tree with the link noise measured in B, it splits 19 to 26% of clusters at the next level. On B, 57.5% of clusters split (Appendix F).

One plausible explanation lies in the difference between visual similarity and linguistic category relations. A label taxonomy organizes categories by meaning, such as a bed being a kind of furniture. Visual concepts may instead connect classes through shape, texture or setting. A class may share its

shape with one set of classes and its setting with another. These relationships need not follow the same hierarchy.

The forest and WordNet differ in how groups of classes nest. Whether the classes grouped together are semantically coherent is a separate question, and WordNet can still answer it for each tree and cluster. Classes within a tree are more similar in appearance and more closely related in WordNet than random classes of the same number. This holds for a separately trained classifier's features, for color and texture statistics, and for the WordNet hierarchy (Appendix F). Peeling assigns every shared concept to a cluster, but that cluster may contain additional classes. To compare the forest with a single tree, we also limit how many additional classes are included. A tree holds a concept when one node contains all its linked classes and no more than twice that number of classes in total. MCP holds a concept when the cluster it is peeled from meets the same bound. One tree built on B by agglomerative clustering holds 62.1% of the 1,227 shared concepts and leaves 465 without such a node. MCP's forest holds 72.6% of the same 1,227 (Appendix G). At this size bound, the forest holds more shared concepts than the single tree tested here. WordNet relates class labels through their meanings. MCP relates classes through the invariants they share. These invariants organize DINOv2's classes into multiple trees rather than one taxonomy. The forest structure of multiple trees with multiple roots lends itself to parallel abduction as discussed in the latter part of this paper.

3.2 Cluster tightness and WordNet coherence

MCP's clusters are tighter than random clusters of the same sizes. The subspace silhouette compares each class's average distance to classes in its own cluster with that to the nearest other cluster (Lipor & Balzano, 2020). The distances use subspaces fitted to the clusters' rows of B, restricted to concepts not yet peeled. Higher scores indicate tighter clusters relative to their separation. The random reference shuffles class assignments while preserving cluster sizes and refits the subspaces after each shuffle. Eleven of the twelve levels qualify. Each has at least ten classes in total and at least two clusters, including one with multiple classes. At all eleven levels, the observed silhouette exceeds the mean of ten shuffled comparisons (Appendix H).

WordNet provides a separate check of whether the classes within a cluster are related in meaning (Miller, 1995; Wu & Palmer, 1994). Wu–Palmer similarity scores two labels by the depth of their closest common ancestor relative to their own depths. Nine of the twelve levels qualify for this comparison, each with at least five clusters containing two or more classes. At every qualifying level, MCP's clusters contain more closely related class labels than size-matched random clusters (Appendix I). The clusters are therefore tighter in concept space and more semantically coherent than random clusters of the same sizes.

4 Invariants in class recognition

Ablating DINOv2's invariants reveals how the model recognizes different classes through class unique and cluster unique concepts. Some classes are recognized mainly through their class unique concepts, others mainly through cluster unique concepts, and the rest through both. For classes recognized through both kinds, removing them together causes a larger average accuracy loss than the sum of the two separate losses.

4.1 Concept removal and cluster-level ablation

Ablation removes the selected concepts' contributions while preserving the rest of the class token. Let x_i be the raw class token of image i, and d_j the decoder vector for concept j. Removing a concept set S gives

$$x_i^{(-S)} = x_i - \sum_{j \in S} z_{i,j} d_j. \tag{3}$$

The contribution of every other concept remains, together with the part of the token the autoencoder does not reconstruct. A linear classifier trained on the raw class tokens of the training images remains frozen during all removals. Before removal, it achieves 89.1% top-1 accuracy on training images and 83.1% on validation images. For each class, we measure the accuracy drop from its own baseline on the same images, in percentage points. Unless stated otherwise, the drops below are measured

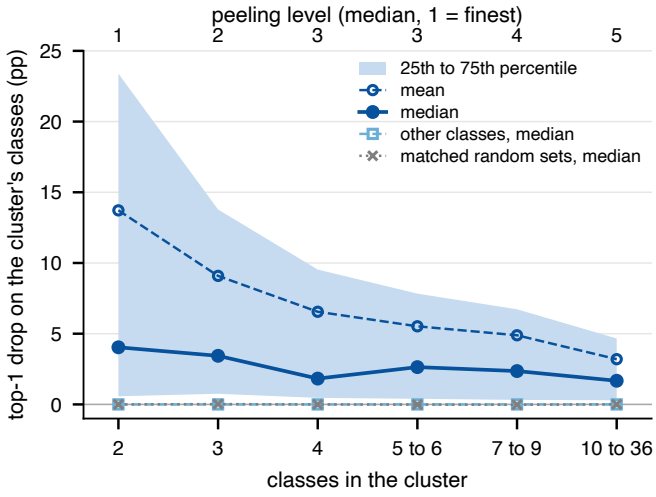


Figure 3: Accuracy loss on a cluster's own classes after removing its peeled concepts, grouped by the number of classes in the cluster. The top axis gives each group's median peeling level. The solid line and band show the median and the 25th to 75th percentiles over clusters, and the dashed line shows the mean. Matched random sets exist for 451 of the 635 clusters (Appendix C.1).

on training images. Classifier training and the comparison with DINOv2's published accuracy are detailed in Appendix A. Matched random removals test whether the selected concepts cause a greater loss than comparable random sets (Appendix C.1).

For class-level ablation, B identifies the class unique and cluster unique concepts linked to each class. These two kinds can be removed separately or together. For cluster-level ablation, MCP identifies the cluster unique concepts peeled from each cluster. Removing this set measures its contribution to recognition across the cluster's classes.

B links each class to its concepts, but it has no clusters or levels. MCP adds both, so the effect of the invariants can be read cluster by cluster. We remove the concepts peeled from one cluster at a time and measure the accuracy loss on that cluster's classes. The loss stays inside the cluster. Classes outside it change by less than 0.01 percentage points at the median. At every level with at least nine clusters, the cluster's own removal also costs more than a matched random removal in 88 to 95% of comparisons (Appendix C.1). Removing the invariants of smaller clusters costs their classes more (Figure 3). Clusters of two classes lose a median of 4.0 percentage points, against 1.7 for clusters of ten or more classes.

4.2 Class unique and cluster unique concepts in recognition

For each class, three separate interventions remove its complete class unique set, its complete cluster unique set or both sets. Among the 872 classes with both kinds, the median accuracy losses are 31.2, 7.8 and 76.3 percentage points, respectively. Removing matched random sets of concepts that have no link to the class barely changes its accuracy (Appendix C.1). The selected concepts cost more than these random sets in 95%, 82% and 100% of the respective comparisons, so the losses come from the concepts themselves. For each of the 872 classes, we subtract the sum of its separate losses from its joint loss. The median excess is 8.5 percentage points. A positive median excess also occurs on DINOv1 (Caron et al., 2021) and DINOv3 (Siméoni et al., 2026) under the same procedure (Appendix C.2, Table 5).

The population medians do not show how the relative effects of the two removals vary across classes. We summarize them for each class using a ratio, the class unique loss fraction u. Let D_{cu} be the accuracy loss after removing all class unique concepts linked to a class. Let D_{sh} be the loss after separately removing all cluster unique concepts linked to that same class. Both losses are measured on the class's own training images. After setting negative losses to zero, we calculate $u = D_{\mathrm{cu}}/(D_{\mathrm{cu}} + D_{\mathrm{sh}})$.

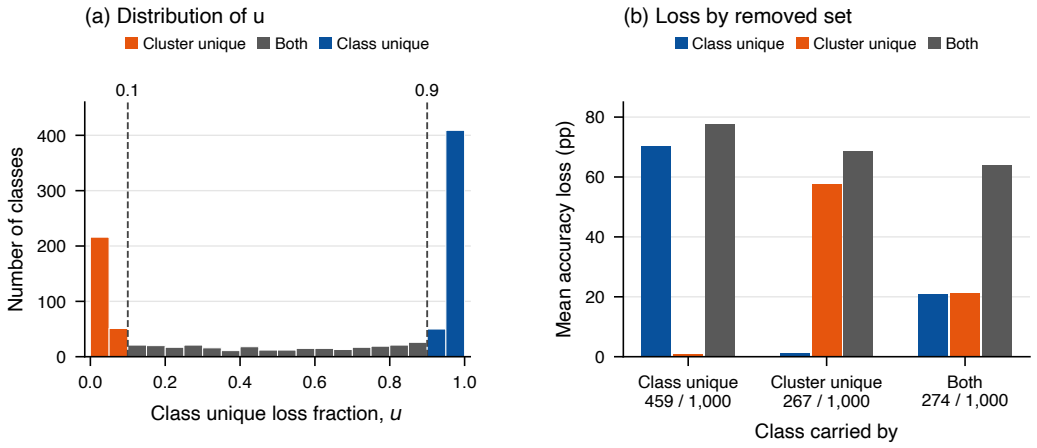


Figure 4: Class unique loss fraction and removal losses on training images. (a) Distribution of u over all 1,000 classes (Section 4.2), with dashed lines at 0.1 and 0.9. (b) Mean accuracy loss in each class group after removing the class unique set, the cluster unique set or both. Exact values are in Table 4 (Appendix C.2).

A value near 1 means that class unique removal causes the larger loss. A value near 0 means that cluster unique removal causes the larger loss. At 0.5, the two separate removals cause equal losses. It should be noted that this ratio does not include the joint removal loss, which is super-additive, as discussed later.

Class unique concepts dominate recognition for 459 of the 1,000 classes, where u is at least 0.9. Their most frequently firing concepts are almost always class unique. Cluster unique concepts dominate for 267 classes, where u is below 0.1. Their most frequently firing concepts are always shared and link to two classes at the median. These classes also belong to larger families of related labels in WordNet. Their validation errors mostly go to related classes. Of these errors, 67% go to a class sharing at least one of their concepts, against 26% for classes with u of at least 0.9. They are also recognized less accurately, at a median validation accuracy of 82% against 92% (Appendix C.2). The 57 classes with no class unique concept are still recognized on 72% of their validation images. Figure 4 shows the distribution of these classes and their removal losses.

Figure 5 illustrates how different concepts contribute to class recognition. For each concept, an attribution map shows the image regions its activation depends on. For grey parrot, the class unique concept's map highlights neck plumage, while the cluster unique concept's map highlights the beak. Removing these concepts individually lowers training accuracy by 34 and less than 1 percentage point, respectively. For snail, the class unique concept's map highlights the shell, while the map of the concept shared with slug highlights the head region in both animals. Removing these concepts costs snail 2 and 70 percentage points, respectively. Fireboat uses both kinds. Its class unique concept's map highlights parts of the water spray and surrounding scene, while its cluster unique concept's map highlights the boat. Removing its complete class unique set, cluster unique set or both costs 40, 26 or 89 percentage points. The attribution maps serve as illustrations (Appendix C.3).

The super-additive effect is strongest among the 274 classes between the two dominance thresholds, where $0.1 \leq u < 0.9$. Removing both kinds costs an average of 21.8 percentage points more than the sum of their separate losses. The corresponding excess is 6.2 points for classes carried by class unique concepts and 9.6 points for classes carried by cluster unique concepts. For the median of the 274 classes, 34% of the images lost under joint removal survive each separate removal (Appendix C.2).

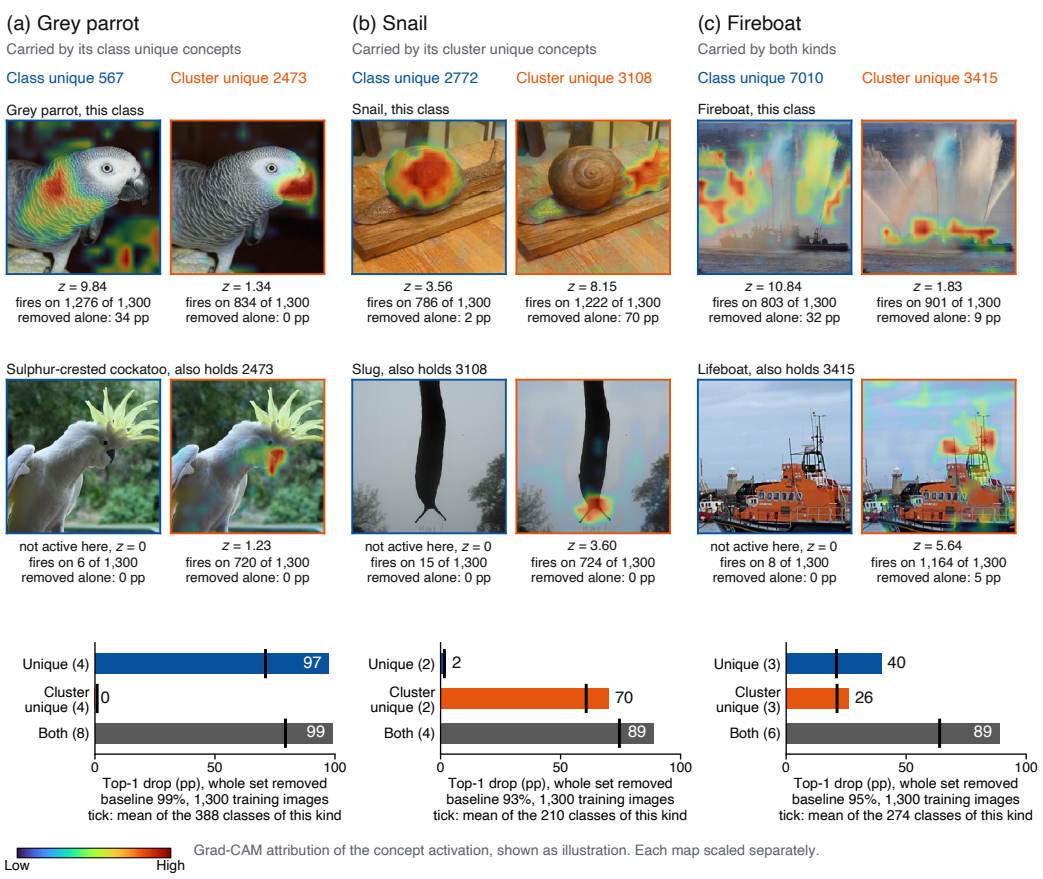


Figure 5: Class unique and cluster unique concepts in three selected cases. Each panel pairs a class (top) with another class that holds the same cluster unique concept (bottom). Under each image, z is the concept's activation, with firing counts and single-concept removal losses on that class's training images. Bars show the accuracy loss after removing whole concept sets, with set sizes in parentheses. Black ticks mark the mean over classes of the same kind that hold both sets. Attribution maps are separately scaled Grad-CAM maps (Selvaraju et al., 2017) shown as illustration (Appendix C.3).

5 Conclusion

MCP turns a sparse autoencoder's concepts into a lens on how a vision model reasons. Read together, the concepts show which classes compete to explain the same image. Four results show how DINOv2 reaches its decision. First, invariants shared by one class or by many reach the decision together. Removing them lowers accuracy mainly on the classes they explain, at every level we test. Second, the super-additive effect of removing class unique and cluster unique concepts together shows that the model weighs both kinds at once when it chooses a class. Either kind alone can keep an image on its true class, and some images are lost only when both are removed. Choosing the class that best explains the image in this way is abduction. Third, classes dominated by cluster unique concepts are mostly misclassified as classes that share those concepts. This holds for 67% of their errors, against 26% when class unique concepts dominate. Fourth, MCP's forest has multiple trees with multiple roots. DINOv2's categories therefore sit in parallel trees rather than one hierarchy searched from coarse to fine. These results suggest that DINOv2 recognizes classes by parallel abduction, considering single classes and clusters of related classes at once. Human vision reaches this decision recursively, as a first glance gives a coarse category and a closer look then settles the details (Hochstein & Ahissar, 2002; Bar, 2003). DINOv2 gets both in one pass, with cluster unique concepts as the coarse category and class unique concepts as the details. The cost is ambiguity, as classes that rely on cluster unique concepts are recognized less accurately. What human vision does over time, the vision transformer does at once.

ETHICS STATEMENT

This study analyzes a frozen public backbone and deploys no decision system. ImageNet's documented problems with offensive labels, non-consensual images and geographic skew can affect the class–concept links and clusters, especially those involving people. The main risk is over-interpretation, mitigated by the scope statements and the near-threshold flags (Appendix D).

REPRODUCIBILITY STATEMENT

Section 2 defines the biadjacency matrix B, the resolution levels and the peeling procedure. Section 4.1 defines concept removal. Appendix A gives the backbone, image preprocessing, autoencoder and classifier settings. Appendix B records the peeling schedule, Appendix C details the ablation procedures and matched controls, and Appendix D describes the matrix stability checks and boundary flags. Appendices E to I report the schedule sensitivity, the nesting tests, the one-tree comparison, the subspace silhouette and the WordNet null. Appendix J reports the model-selection comparisons across token substrates and dictionary sizes. Appendix K gives the limitations in full and the release details. The containment forest of Section 3.1 is browsable at local-structure-explorer.vercel.app.

AI USE STATEMENT

We used generative AI tools to draft and revise manuscript text, improve its clarity and organization, and assist with literature retrieval and discovery. We also used these tools to prepare figures, check and revise analysis code, and assist with interpreting experimental results. The authors take responsibility for the final manuscript, analyses and conclusions.

REFERENCES

Moshe Bar. A cortical mechanism for triggering top-down facilitation in visual object recognition. *Journal of Cognitive Neuroscience*, 15(4):600–609, 2003. doi: 10.1162/089892903321662976.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. arXiv:1704.05796.

Michał Brzozowski and Neo Christopher Chung. Ablating archetypes: The stability of archetypal SAEs is an artifact of initialization and metric design. *arXiv:2606.02061*, 2026.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with Matryoshka sparse autoencoders. In *ICML*, 2025. arXiv:2503.17547.

Tue M. Cao, Hoang X. Nhat, Raed Alharbi, Phi Le Nguyen, and My T. Thai. Tree SAE: Learning hierarchical feature structures in sparse autoencoders. *arXiv:2605.07922*, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. arXiv:2104.14294.

Anil Damle, Victor Minden, and Lexing Ying. Simple, direct and efficient multi-way spectral clustering. *Information and Inference*, 8(1):181–203, 2019.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. arXiv:2309.16588.

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 *ICLR*, 2021. arXiv:2010.11929.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept recursive activation factorization for explainability. In *CVPR*, 2023. arXiv:2211.10154.

Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *ICML*, 2025. arXiv:2502.12892.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *ICLR*, 2025. arXiv:2406.04093.

Gleb Gerasimov, Timofei Rusalev, Nikita Balagansky, Daniil Laptev, Vadim Kurochkin, and Daniil Gavrilov. Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders. *arXiv:2606.12138*, 2026.

Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. In *NeurIPS*, 2019. arXiv:1902.03129.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. In *NeurIPS*, 2025. arXiv:2503.01822.

Shaul Hochstein and Merav Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804, 2002. doi: 10.1016/S0896-6273(02)01091-7.

Sonia Joseph, Praneet Suresh, Ethan Goldfarb, Lorenz Hufe, Yossi Gandelsman, Robert Graham, Danilo Bzdok, Wojciech Samek, and Blake Aaron Richards. Steering CLIP's vision transformer with sparse autoencoders. In *CVPR Workshops*, 2025. arXiv:2504.08729.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *ICML*, 2018. arXiv:1711.11279.

Matthew Kowal, Richard P. Wildes, and Konstantinos G. Derpanis. Visual concept connectome (VCC): Open world concept discovery and their interlayer connections in deep models. In *CVPR*, 2024. arXiv:2404.02233.

John Lipor and Laura Balzano. Clustering quality metrics for subspace clustering. *Pattern Recognition*, 104: 107328, 2020.

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

Nghia Nguyen, Tianjiao Ding, and René Vidal. Hierarchical concept embedding & pursuit for interpretable image classification. In *CVPR*, 2026. arXiv:2602.11448.

Matthew Lyle Olson, Musashi Hinck, Neale Ratzlaff, Changbai Li, Phillip Howard, Vasudev Lal, and Shao-Yen Tseng. Analyzing hierarchical structure in vision models with sparse autoencoders. In *CVPR Workshops*, 2025. arXiv:2505.15970.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. arXiv:2304.07193.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *NeurIPS*, 2025. arXiv:2504.02821.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv:2501.16615*, 2025.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B*, 78(5):947–1012, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. arXiv:1610.02391.

Thomas Serre. *Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines*. PhD thesis, Massachusetts Institute of Technology, 2006.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. arXiv:2508.10104.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders. *arXiv:2502.06755*, 2025.

Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *ACL*, pp. 133–138, 1994.

Yusong Zhao, Hengyi Wang, Tanuja Ganu, Akshay Nambi, and Hao Wang. Cascaded sparse autoencoders learn multi-level visual concepts in multimodal LLMs. *arXiv:2606.16193*, 2026.

Appendix

A SETUP CONSTANTS

Table 1 collects the constants behind Sections 2 to 4.

Table 1: Setup constants. Normalized mean squared error (NMSE) is $\sum_i (y_i - \hat{y}_i)^2 / \sum_i y_i^2$. Accuracy differences are reported in percentage points (pp). A dead concept fires on no training image.

Backbone and token	DINOv2 ViT-B/14 with registers (Darcet et al., 2024), class token after final layer normalization, 768 dimensions
Images	1,281,167 training and 50,000 validation images from ImageNet-1k, bicubic resize to 256, center crop to 224
Autoencoder	Archetypal architecture, 8,192 concepts, top-eight selection, mean-centered, archetype-initialized, 50 epochs
Autoencoder training	Adam at 4×10^{-4}, cosine decay, batch size 4,096, seeds 42 and 43, trained on the class tokens of the 1,281,167 training images
Seed 42 quality	NMSE 0.229 on training and 0.240 on validation, 11.5% dead concepts
Seed 43 quality	NMSE 0.228 on training and 0.240 on validation, 10.5% dead concepts
Linear classifier	Top-1 accuracy 89.1% on training and 83.1% on validation
Matrix B	Firing rate ≥ 0.2, 4,279 active concepts, 6,299 positive links, 1,227 shared concepts, 14.0% of positive links boundary-flagged (Appendix D)
Clustering	SSC Lasso α=3×10^{-6}, spectral step of Damle et al. (2019)
Peeling rule	A concept is peeled when its remaining linked classes all fall within one cluster
Ablation identity	Masked and subtracted removal differ by 8.9×10^{-16} on the token and 4.3×10^{-14} on the logits, zero predicted-class changes
Matched controls	Up to 20 per removed set, each matched quantity within $\pm25\%$. Class level draws concepts with no link to the class. It matches the number of the class's images they fire on and, when possible, how much of the class token they remove. Cluster level matches activation mass and the number of fired images over all training images
WordNet null	25 size-matched random-cluster draws per level for mean medoid distance and least common subsumer similarity

Training seeds. The archetypal architecture is reported to remove the run-to-run instability of autoencoder features (Fel et al., 2025). A later analysis traces that stability to a k-means initialization shared across runs (Brzozowski & Chung, 2026). We therefore train our two autoencoders from independent initializations. The training seed is the autoencoder's and sets both the archetype initialization and the training run. The clustering step keeps one fixed seed in every run. The peel and the causal tests on the second seed are reported in Appendices B and C.2.

Linear classifier and reference accuracy. The classifier maps the final 768-dimensional class token to 1,000 class scores. It is trained on all 1,281,167 ImageNet training images with cross-entropy loss. Training uses SGD for ten epochs with batch size 1,024 and momentum 0.9, without weight decay. The learning rate decays to zero with a cosine schedule. The class tokens are not length-normalized. Six initial learning rates from 0.0005 to 0.1 are tested. The rate of 0.1 gives the highest validation accuracy. Its best epoch is the last of the ten. Because both choices use validation accuracy, the validation figure is not a held-out estimate. The two best rates differ by 0.02 pp, so this choice barely changes it. The classifier reaches 89.13% training and 83.07% validation top-1 accuracy. It stays frozen during every removal, so each loss compares accuracy before and after removal under the same classifier.

The official DINOv2 repository reports 84.6% ImageNet linear accuracy for ViT-B/14 with registers, about 1.5 pp above our classifier.[1] The released classifiers combine class tokens with averaged patch

[1] `https://github.com/facebookresearch/dinov2`, pretrained-model results, accessed September 18, 2026.

tokens. One variant uses the final layer, and another uses class tokens from the last four layers. Our classifier uses only the final class token. The autoencoder is trained on the same token. The published accuracy therefore provides a reference under a different feature configuration.

Other backbones. The replication in Appendix C.2 uses DINOv1 ViT-B/16 (Caron et al., 2021) and DINOv3 ViT-B/16 with four registers (Siméoni et al., 2026). Both backbones give the class token after the final layer normalization, with 768 dimensions. Images are preprocessed as for DINOv2. Before autoencoder training, each backbone's features are multiplied by one scalar. The scalar is 0.401 for DINOv1 and 1.562 for DINOv3. It matches their mean centered norm to that of DINOv2. The recipe bounds how far each dictionary element may move from its archetypes and fixes the learning rate, both in absolute units. The scalar keeps these settings at the feature scale where the recipe was set. All other constants follow Table 1. Table 5 gives the dictionary sizes and seeds.

B PEELING SCHEDULES

B.1 LEDGERS AND COVERAGE

We call the matrix built from all training images the training-derived matrix. The matrix built from 50 validation images per class is the validation-derived matrix. Table 2 gives both peeling schedules. Level 0 removes the class unique concepts of every active class without clustering. The classes that still hold shared concepts enter level 1. At each later level l the schedule clusters the a_l classes that still have concepts. The target size s_l is the FSS value the level visits, and $K_l = \lfloor a_l / s_l \rceil$ with round half to even and a floor of 1. The peeled columns count class unique concepts at level 0 and shared concepts after it. Both schedules end at the level where $K_l = 1$, since a single cluster peels every remaining concept. This is level 12 on the training-derived matrix and level 13 on the validation-derived matrix. The training-derived matrix also holds one concept with FSS 36. It is peeled at level 7, and no level with $s_l = 36$ runs because K_l reaches 1 at $s_l = 20$.

While at least two clusters exist, the training-derived schedule peels 1,226 of its 1,227 shared concepts. The validation-derived schedule peels 1,540 of 1,541. The single-cluster level then peels the remaining concept by construction. Rebuilding the matrix from other images or with another autoencoder also leaves at most four shared concepts unpeeled while at least two clusters exist. On its validation-derived matrix, the seed 43 autoencoder peels 1,562 of 1,563. A matrix built the same way from 50 training images per class peels 1,472 of 1,474. Two further matrices come from disjoint halves of the training images. Each half holds 640.6 images per class on average and at most 650. They peel 1,249 of 1,253 and 1,248 of 1,248. An autoencoder trained on one half, with its matrix built from the other half, peels 1,186 of 1,188.

Table 2: Peeling schedule ledgers. l is the level. s_l is the FSS value the level visits and a_l counts the classes clustered. K_l is the cluster count and the peeled columns count concepts removed at each level. Level 0 removes class unique concepts without clustering, so its a_l counts all active classes.

	training-derived matrix				validation-derived matrix			
l	s_l	a_l	K_l	peeled	s_l	a_l	K_l	peeled
0		1,000		3,052		999		2,896
1	2	929	464	482	2	971	486	533
2	3	814	271	284	3	901	300	282
3	4	728	182	219	4	861	215	181
4	5	575	115	112	5	810	162	180
5	6	408	68	58	6	719	120	136
6	7	287	41	32	7	566	81	108
7	8	204	26	20	8	415	52	51
8	9	111	12	15	9	283	31	38
9	10	47	5	2	11	201	18	15
10	12	35	3	0	16	147	9	12
11	17	35	2	2	19	92	5	2
12	20	12	1	1	27	80	3	2
13					35	35	1	1

B.2 MOTIVATION FOR PEELING

Section 2.3 gives three reasons for peeling. The paragraphs below give the evidence for each.

The first reason is that each level can look for invariants shared by a different number of classes. Peeling removes a concept from B once its classes fall into one cluster. On the training-derived matrix, 537 of the 1,227 shared concepts are peeled at the level whose target size equals their FSS. Another 660 are peeled at a later level and 30 at an earlier one. So 1,197 concepts are peeled once the target size reaches their FSS.

The second reason is that small invariants act as noise for larger ones. Without peeling, every level would cut the same adjacency, and the small invariants that dominate it would lead the clustering at every target size. A coarser cut of the same adjacency mostly merges the clusters that the small invariants form. A merged cluster need not hold together the classes of a larger invariant. Peeling rebuilds the adjacency from the links that no earlier level has explained, so a larger target size can group classes by a broader invariant.

To test this, we run the training-derived schedule without peeling. Level 0 removes the class unique concepts as before. Every later level clusters the same 929 classes over all 1,227 shared concepts, with $K_l = \lfloor 929/s_l \rceil$, so the adjacency is built once. The first level is therefore identical to level 1 of the peeling schedule. A cluster splits when its classes fall into several clusters at the next level. Without peeling, 380 of 1,826 clusters split, against 506 of 880 with peeling (Appendix F). The coarser levels therefore mostly merge finer clusters. These merges often miss the larger invariants. For 141 of the 412 concepts with FSS 3 to 5, the first cluster holding all their classes is at least five times their FSS. With peeling, this holds for 14. Concepts with FSS 2 are rarely affected, with 40 of 786 against 26. The coarser levels also keep some small clusters and merge the remaining classes into clusters of up to 97 classes at $s_l = 20$.

Two further comparisons test the noise itself. They change only the concept columns and keep the classes and cluster count fixed within each comparison. In the first, each level from 2 to 11 keeps its classes and K_l, and the concepts peeled at earlier levels are put back. The number of concepts peeled at these levels falls from 744 to 489. Take the 411 concepts whose FSS equals the level's target size. With peeling, 71 of them have all their classes in one cluster. With the peeled concepts put back, 50 do (exact McNemar test, $p = 0.038$). Level 2 shows no gain, with 29 against 30. In the second, each target size clusters the classes that hold a concept with FSS at least s_l. It clusters them once with all shared concepts and once without those whose FSS is below s_l. Each of the 441 concepts with FSS 3 or more is tested at the target size equal to its FSS. The number whose classes fall into one cluster rises from 96 to 141 without the smaller concepts (exact McNemar test, $p < 0.001$). This comparison removes every smaller concept, while peeling removes only those that earlier levels have already peeled. At $s_l = 5$, 180 of the 242 remaining concepts still have FSS below 5. In both comparisons, the arm without the smaller concepts also has fewer concept dimensions. These tests therefore do not separate noise from the third reason in Section 2.3.

The third reason is that peeling reduces the number of concept dimensions. Each remaining class is a data point for the clustering, and each remaining shared concept is a dimension. The ratio of classes to remaining shared concepts is 0.8 at level 1, with 929 classes and 1,227 concepts. It rises to 5.6 at level 8, with 111 classes and 20 concepts (Table 2).

C ABLATION DETAILS

This appendix gives the removal procedure and its controls in C.1. The class-level ablation and the case studies follow in C.2 and C.3.

C.1 CONCEPT REMOVAL AND MATCHED CONTROLS

Removal from the raw class token. Equation 3 preserves the reconstruction error and removes only the selected concepts' decoder contributions. With classifier weights W and bias b, the class scores after removal are

$$Wx_i^{(-S)} + b = Wx_i + b - \sum_{j \in S} z_{i,j} W d_j. \tag{4}$$

This identity allows the removal to be evaluated directly on the class scores. For each class, the loss is its top-1 accuracy before removal minus its accuracy after removal on the same images, in percentage points.

Controls for class-level removal. Section 4.2 asks whether a removal loss comes from the concepts that B links to a class. A removal might also lower accuracy simply by disturbing the class token on the class's images. A useful control must therefore act on the same images as the removed set. We build each control from concepts that have no link to the class in B. Concepts are drawn in random order until the set fires on at least 75% as many of the class's training images as the removed set. The set is kept if it fires on at most 125% as many. These are the image-matched controls.

Matching the images is not enough on its own. An image-matched control still removes much less of the class token than the removed set. We measure this with the removal norm. It is the mean norm of the part of the class token that a set removes, over the class's images. The median image-matched control has 0.33, 0.38 and 0.21 times the removal norm of the removed set for class unique, cluster unique and joint removal. We therefore also build norm-matched controls, whose removal norm lies within 25% of that of the removed set. For these controls, concepts that carry more of the class token on the class's images are more likely to be drawn first. Each removal receives up to twenty controls of each type from at most 200 draws.

Norm-matched controls are harder to find. A class counts as covered when at least five are found for it. We expected at least 70% of the classes to be covered in class unique and cluster unique removal. Only 52% and 60% were covered, and 18% under joint removal. We use norm-matched controls for an intervention when most of its classes are covered. Class unique and cluster unique removal are therefore compared with norm-matched controls, and joint removal with image-matched ones. The covered classes are not typical, however. Their own concepts cost less to remove. Class unique removal costs them a median of 8.7 percentage points, against 35.2 over all 943 classes. For cluster unique removal, the two medians are 1.6 and 9.6 points.

Neither kind of control leaves accuracy exactly unchanged. Almost every random removal moves the class's accuracy a little, up or down, and the changes largely cancel. Image-matched controls have median losses from -0.6 to -0.4 percentage points. Negative losses mean that accuracy rose. Norm-matched controls have median losses from -0.2 to 0.4. The selected concepts cost more than image-matched controls in 98%, 93% and 100% of comparisons. Against norm-matched controls, they cost more in 95% and 82% of comparisons for class unique and cluster unique removal. For joint removal, they cost more in over 99.9% of comparisons on the covered classes. The loss therefore depends on which concepts are removed, even when a random removal disturbs the same images by a similar amount.

Controls for cluster-level removal. Section 4.1 compares each cluster's removal with the same kind of norm-matched controls. They are built on the training images of the cluster's classes, from concepts with no link to any of those classes. At least five are found for 451 of the 635 clusters. Their median loss lies between -0.03 and 0.06 percentage points at every level with at least nine clusters. The cluster's own removal costs more in 88 to 95% of the comparisons at each of these levels.

Earlier controls for cluster-level removal. The comparison in Appendix I uses an earlier rule. Each cluster's removed set is compared with up to twenty random sets drawn from the 4,279 concepts active in B. Each control contains the same number of concepts as the removed set. Two further quantities are matched within $\pm 25\%$, using the training images. The first is decoder-weighted activation mass, the sum of $z_{i,j}^2 \|d_j\|^2$ over all training images and selected concepts. The second is the number of images on which at least one selected concept fires. Sampling stops after twenty accepted controls or 3,000 attempts. Accuracy losses and the fraction of comparisons in which the real set causes a larger loss are reported separately.

Identity check. Masking a concept's code and subtracting its decoder contribution give numerically equal tokens after removal. The check covers 30 removed sets on 1,000 random training images, on which the removed sets fire 99 times. Across these removals, the maximum difference is 8.9×10^{-16} on the token and 4.3×10^{-14} on the logits. The executed pipeline stores codes as 16-bit floats and computes in 32 bits. It differs from a 64-bit reference by at most 5.3×10^{-3} in its logits, with zero changes in the predicted class.

C.2 CLASS-LEVEL ABLATION

Class groups before removal. Table 3 describes the class groups of Section 4.2 before any concept is removed. The 57 classes with no class unique concept are recognized at a median 79.4% of their training images and 72.0% of their validation images. On training images their interquartile range is 65.0 to 84.4, and their tenth percentile is 53.5. The 71 classes with no shared concept reach 93.2% and 88.0%, with an interquartile range of 86.7 to 96.2. A one-sided Mann–Whitney test between the two groups gives $p = 9 \times 10^{-13}$ on training images. On the second autoencoder seed the groups hold 54 and 55 classes and sit at 80.2% and 90.9% (72.0% and 82.0% on validation). The same contrast appears under intervention. We keep only a class's class unique concepts, removing every other concept and keeping the reconstruction residual. This retains a median 93.5% of baseline accuracy on training images over the 943 classes that have such concepts. Keeping only the shared concepts retains 66.5% over the 929 classes that have them. On validation images the two figures are 94.1% and 66.7%. The lowest training accuracies among the 57 are 43% for tights, 50% for water jug and tape player, and 52% for husky and CRT screen.

The characteristics in Table 3 hold beyond this seed and backbone. On the second seed, classes where cluster unique concepts dominate sit in larger WordNet families than classes where class unique concepts dominate. The family medians are 10.5 against 8 classes within Wu–Palmer similarity 0.8. Their most active shared concept links two classes at the median. On DINOv1 and DINOv3 the family medians are 10 against 8. The most active shared concept links a median of 3 classes on DINOv1 and 2 on DINOv3. The animal share is higher where cluster unique concepts dominate on DINOv2 (58% against 34%) and DINOv1 (50% against 36%). It is similar on DINOv3 (40% against 42%). Group sizes differ across backbones (566 and 144 classes on DINOv3, 415 and 406 on DINOv1), so the table reports characteristics per group rather than counts.

Table 3: Class groups of Section 4.2 before removal, on the main autoencoder. Train and Val give the median top-1 accuracy of the raw linear classifier over the group's classes, in %. Own and Shared give the median number of class unique and shared concepts per class in B. Top FSS is the median number of classes linked by the class's most active shared concept, and Level ≤ 2 the share of those concepts peeled at the first two levels. Family is the number of other classes within Wu–Palmer similarity 0.8 of the class label, median and mean. Animal is the share of classes under the WordNet animal branch. The last two rows are the classes at the ends of the u scale.

Group	Classes	Train	Val	Own	Shared	Top FSS	Level ≤ 2	Family	Animal
Class unique dominates ($u \geq 0.9$)	459	94.2	92.0	4	2	3	41%	8 / 12.1	34%
Both kinds ($0.1 \leq u < 0.9$)	274	88.0	80.0	3	3	2	54%	7 / 11.8	31%
Cluster unique dominates ($u < 0.1$)	267	88.5	82.0	2	4	2	62%	11 / 19.4	58%
No shared concept	71	93.2	88.0	4	0	–	–	6 / 8.5	6%
No class unique concept	57	79.4	72.0	0	4	3	53%	11 / 19.4	37%

Table 4: Mean accuracy losses in percentage points across each class group. Unique, Shared and Both remove the complete class unique set, shared set, or both. Excess is the joint loss minus the two separate losses, calculated per class before averaging. Rows correspond to $u \geq 0.9$, $u < 0.1$ and $0.1 \leq u < 0.9$. Counts cover all 1,000 classes.

Carried by	Classes	Unique	Shared	Both	Excess
Class unique	459	70.4	0.8	77.5	6.2
Shared	267	1.2	57.7	68.6	9.6
Both	274	20.9	21.2	63.9	21.8

Replication on DINOv1 and DINOv3. The class-level ablation of Section 4.2 was repeated on two other DINO backbones with the same procedure. Each backbone has its own autoencoder and a training-derived matrix B at the same firing-rate threshold. The peeling schedule is the same, and so are the removals from the raw class token with the raw linear classifier. On each DINOv1 and DINOv3 run in Table 5, all but two to five shared concepts peel while at least two clusters exist. Table 5 gives the training-set medians over classes that hold both kinds. On every backbone, removing the class unique set costs more than removing the shared set. The joint removal exceeds the sum

of the two separate losses by 5.6 to 9.2 percentage points. The two losses substitute across classes, with Spearman correlations between -0.79 and -0.92. Removing a cluster's peeled set at the finest level costs its classes 6.6 to 15.5 points and other classes at most 0.1. The DINOv3 dictionary of 8,192 concepts keeps only 582 shared concepts and shows the weakest substitution. Its 4,096-concept dictionaries match the other backbones. DINOv1 and DINOv3 features are each rescaled by one scalar before autoencoder training (Appendix A). On DINOv2, a second autoencoder seed repeats the class-level ablation with nearly identical medians (Table 5). The scope of the two-kind finding is therefore three DINO backbones under one recipe, not vision transformers in general.

Table 5: Class-level and cluster-level ablation on three DINO backbones. Probe is the top-1 accuracy of the raw linear classifier on validation images, in %. Unique and Shared under the first heading count concepts in B. Unique, Shared and Both under the second heading are median accuracy losses in percentage points on training images after removing the complete class unique set, shared set, or both, over classes holding both kinds. Excess is the median of the per-class joint loss minus the two separate losses. Spearman is the rank correlation between the two separate losses across those classes. Cluster gives the median loss within a cluster at the finest peeling level and the largest median change outside any cluster over levels. The 4k and 8k labels give the dictionary size. A dash marks a result not computed for that run.

Backbone, dictionary	Probe	Unique	Shared	Unique	Shared	Both	Excess	Spearman	Cluster
DINOv2, 8k (main text)	83.1	3,052	1,227	31.2	7.8	76.3	8.5	-0.91	7.2 / 0.0
DINOv2, 8k, seed 43	83.1	3,019	1,283	31.2	7.5	76.8	8.5	-0.91	–
DINOv2, 4k	83.1	1,793	1,618	37.4	8.0	80.0	8.5	-0.92	8.2 / 0.0
DINOv1, 4k	77.0	1,322	1,018	42.9	4.9	77.3	6.0	-0.91	15.5 / 0.0
DINOv1, 4k, seed 43	77.0	1,370	1,033	45.3	5.6	76.4	5.6	-0.91	14.6 / 0.1
DINOv3, 4k	83.3	2,268	1,086	51.5	3.6	74.5	9.0	-0.87	8.4 / 0.0
DINOv3, 4k, seed 43	83.3	2,279	1,076	49.8	3.9	74.4	9.2	-0.89	6.6 / 0.0
DINOv3, 8k	83.3	3,595	582	42.3	2.2	64.9	7.2	-0.79	6.9 / 0.0

Class neighbours and removal patterns. Classes with larger relative losses from shared-concept removal have more similar neighbours in class-token space. For each class, we average its training-image tokens and subtract the feature mean used by the autoencoder. Its nearest neighbour is the other class with the highest cosine similarity between these centered means. Among classes with $u < 0.1$, the median nearest-neighbour similarity is 0.73. Among classes with $u \geq 0.9$, it is 0.39. The neighbour shares a concept with the class in B for 97% and 56% of these two groups, respectively.

Class-token similarity separates these removal patterns more strongly than WordNet similarity. The area under the receiver operating characteristic curve measures how well a score separates the two groups. Nearest-class cosine similarity achieves 0.87. The corresponding value is 0.66 for the neighbour's Wu–Palmer similarity and 0.60 for the closest label in WordNet.

Accuracy tracks the nearest neighbour more than the kind. Across the 1,000 classes, the rank correlation between validation accuracy and u is 0.31. Between accuracy and the nearest-neighbour similarity defined above it is -0.48. Holding the neighbour similarity fixed, the partial rank correlation between accuracy and u is 0.01 (95% interval -0.05 to 0.07). Holding u fixed, the partial correlation with the neighbour similarity is -0.39. We also split the classes into fifths by neighbour similarity. Within each fifth the two dominant groups have similar mean accuracy, except in the closest fifth. There, 112 classes dominated by cluster unique concepts average 70% and 9 classes dominated by class unique concepts average 76%. Reliance on shared concepts and lower accuracy are two signs of the same structure. These statistics do not say which produces which.

Where errors go. We take the validation images that the classifier misclassifies before any removal. We ask whether the predicted class shares at least one concept with the true class in B. Errors on classes that share no concept with any other class are excluded, which leaves 7,932 errors. For classes with $u < 0.1$, 66.6% of errors go to a class sharing a concept, 78 times the rate expected if errors went to random other classes. For classes with $u \geq 0.9$, 26.2% do, 48 times that rate. Classes differ in how many other classes share their concepts, so we split them into four groups by that number. The difference between the two kinds of class holds in every group, with a weighted mean of 33

percentage points. Among classes with at least three errors, the median per-class fractions are 70% and 20% (one-sided Mann–Whitney $p < 10^{-33}$).

Joint-removal excess and errors on individual images. The largest average joint-removal excess occurs among classes with contributions from both kinds. For each class, the excess is its joint loss minus the sum of its two separate losses. Among the 274 classes with $0.1 \leq u < 0.9$, the separate losses average 20.9 and 21.2 percentage points. Their joint loss averages 63.9 percentage points, giving a mean excess of 21.8 points. The mean excess is 6.2 points for classes with $u \geq 0.9$ and 9.6 points for classes with $u < 0.1$ (Table 4). The larger excess thus occurs where both kinds contribute individually to recognition.

The two separate removals also affect different images within a class. For each of these 274 classes, we consider training images classified correctly before removal. We identify the images misclassified after class unique removal and those misclassified after shared removal. Their intersection contains a median of 31% of the smaller set, calculated across classes. The median ratio of their intersection to their union is 0.075. Joint removal also creates errors on images that survive either separate removal. For each class, we calculate what fraction of its newly misclassified images under joint removal survived both separate tests. The median fraction is 34%. These images retain enough evidence for recognition after either separate removal, but lose it when both kinds are removed.

Geometric control for the separate removal losses. The larger joint loss persists when a geometric control is fitted to reproduce the separate losses. This control removes components from individual image tokens along two directions derived from centered class means. One follows the nearest class's mean. The other follows the component of the target class's mean perpendicular to that direction. We fit the two removal strengths on one half of each class's training images and evaluate them on the other half. Across the 274 classes, the mean absolute differences from the two separate concept-removal losses are 2.1 and 1.7 percentage points. Joint concept removal nevertheless causes a mean loss 11.4 points greater than joint geometric removal. The 95% confidence interval is 9.7 to 12.9 points. Matching the separate losses therefore leaves a larger joint loss for the concept sets.

C.3 Case studies

Attribution maps in the case studies. The attribution maps in Figure 5 are gradient-weighted class activation maps (Selvaraju et al., 2017). The target is the activation of one concept on one image. The features are the patch tokens that enter the final attention block, and we keep the positive part of the map. The autoencoder reads one class token per image, so a concept has one activation per image and no spatial activation of its own. A map attributes that single number to image regions after the fact. We therefore use the maps as illustration and rest no claim on them. On nine image and concept pairs, blurring the tenth of the image that a map ranks highest lowers the activation by a median of 4%.

Two checks on a random sample show what the maps do carry. We draw 150 classes at random from the 872 classes that hold both a class unique and a shared concept. The draw takes 50 from each of the three kinds of class in Table 4. For each class we take the class unique concept and the shared concept that fire most often on it. We then take 20 random training images on which both fire. The first check concerns consistency across images. The method weights each feature channel by its gradient averaged over positions. For one concept, this weight vector has a median pairwise cosine of 0.81 across images. Of the 300 concepts, 88% reach 0.7. The weights are largely a property of the concept and not of the image. Replacing an image's own vector by the mean of the other images reproduces its map with a median correlation of 0.90. A map therefore mostly shows where one direction in feature space is present, and that direction belongs to the concept. We had predicted a median cosine between 0.65 and 0.80. The sample came out slightly above that range. The second check concerns specificity. On the same image, the map of a concept and the map of a randomly drawn concept have a median cosine of 0.24 over 6,000 comparisons. The maps depend on the concept and are not a saliency map of the image.

The three classes in the case studies were chosen and not sampled. We kept the classes whose two leading concepts fire on at least half of their training images. We also required that the shared concept fires on at least half of the images of another class. Among these we kept the classes whose ablation losses show their kind clearly, which left 36 classes. From these we chose three whose maps were easiest to read, and one image each among 20 random ones. The lower of the two cosines is 0.86,

0.64 and 0.76 for the three classes. The snail's shared concept therefore falls below the 0.7 mark. The cases illustrate the three kinds of class and do not estimate how common such clear cases are. The numbers in the previous paragraph come from the random sample only.

D MATRIX STABILITY AND CALIBRATION

This appendix tests how much of B depends on which images build it and where its threshold sits.

Sampling stability. For the sample-size sweep we permute each class's images once with a fixed seed. At target cap n, each of two disjoint sides uses $\min\{n, \lfloor n_c/2 \rfloor\}$ images of class c, so cap 650 averages 640.6 images per side. Table 6 gives the entry Jaccard similarity between matrices built from the two sides. A counting model with no adjustable parameter assumes each concept fires independently at its full-training rate and applies the same threshold to the sampled count. Its predictions sit within 0.006 of the observations at every tested n. Figure 6 plots the sweep alongside the counting model, and the shared-concept count settles toward the 1,227 of the training-derived matrix as n grows. The observed instability is therefore consistent with the sampling variability of a thresholded count. This test does not measure dictionary stability across independently trained autoencoders.

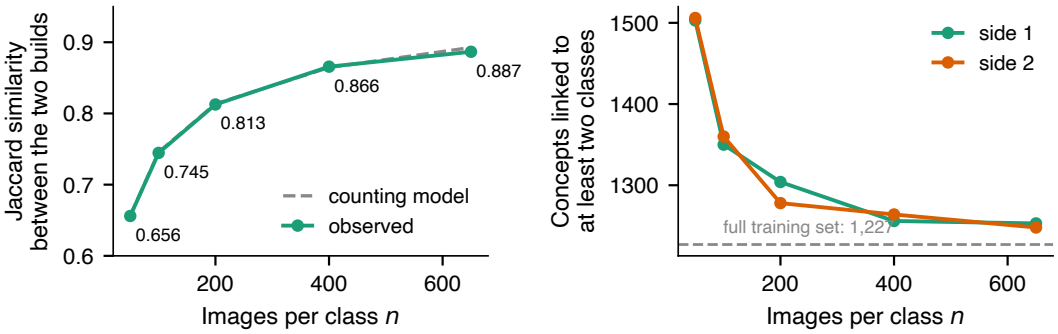


Figure 6: Matrix stability increases with sample size. (a) Entry Jaccard similarity between two disjoint matrix builds at each per-class image cap, observed (teal) and predicted by the counting model (gray dashed). (b) Number of concepts linked to at least two classes in each build, per side. The dashed line marks the 1,227 shared concepts of the training-derived matrix. Both builds use the 0.2 firing-rate threshold.

Boundary flags and count thresholds. Links within two standard errors of the rate threshold carry a boundary flag. These are 880 links, or 14.0% of the positive links, in the training-derived matrix. The fraction is 53.9% in the validation-derived matrix with its 50 images per class. In an $n = 50$ sweep of count thresholds, agreement rises with the threshold (Table 6). Under a class-independence null, the expected number of false links falls from 3.8×10^5 at threshold 1 to 0.002 at threshold 10.

Rate threshold. Raising the threshold above 0.2 improves split-half agreement but strands classes without any link. Table 6 lists the sweep. Agreement moves from 0.861 to 0.928 across it. The threshold 0.2 is the largest tested value at which every class keeps at least one linked concept.

Unseen images and an independent collection. An autoencoder trained on one half of the training images gives Jaccard 0.892 between matrices built from its training half and the unseen half. The seed 42 autoencoder gives 0.887 between the same halves. The half-trained autoencoder's per-concept firing rates correlate at Pearson 0.9932 between the halves, with a mean seen-minus-unseen shift of $+0.0018$. The shift for the seed 42 autoencoder is $+0.0003$. On ImageNetV2 the frozen linear classifier reaches 75.2% top-1 accuracy. Across positive links, training and ImageNetV2 firing rates correlate at Pearson 0.758 before and 0.882 after correction for split-half reliability. The reliability estimate is 0.740 for 10 images per class. Matrices built from 10 images per class have mean entry Jaccard 0.347 between disjoint training samples, 0.334 between training and validation, and 0.307 between training and ImageNetV2.

Table 6: Calibration sweeps. Top, entry Jaccard similarity between two disjoint image samples at each per-class cap n, observed and predicted by the counting model. Middle, the same similarity at $n = 50$ under different count thresholds. Bottom, the firing-rate threshold sweep with the split-half Jaccard similarity, the number of shared concepts retained and the number of classes left without a link.

cap n	50	100	200	400	650			
observed	0.656	0.745	0.813	0.866	0.887			
counting model	0.657	0.746	0.813	0.864	0.893			
count threshold	1	3	5	7	10	15		
Jaccard	0.293	0.514	0.579	0.618	0.659	0.695		
rate threshold	0.1	0.15	0.2	0.225	0.25	0.3	0.4	0.5
Jaccard	0.861	0.884	0.887	0.894	0.898	0.911	0.911	0.928
shared concepts	3,280	2,011	1,227	961	760	503	246	164
uncovered classes	0	0	0	1	3	4	29	82

Firing rate and function. Concept influence is graded across the threshold. The target-class drop from removing one concept rises with firing rate, at Spearman 0.52 over the 6,299 links, with no visible discontinuity at 0.2 (Figure 7a). The concepts below the threshold are the 2,185 concepts outside B whose highest class firing rate lies between 0.05 and 0.2. Removing all of them at once costs 2.32 pp of top-1 accuracy on the training images, while random concept sets with the same activation mass cost 8.08 pp (Figure 7b). The threshold is therefore a coverage choice. It keeps every class linked, and what it leaves out carries little function.

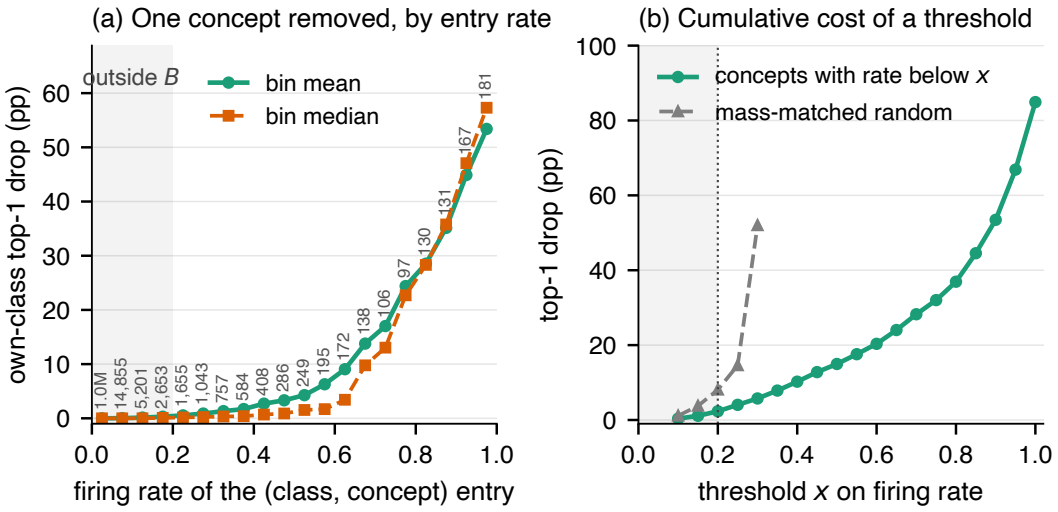


Figure 7: Ablation effects vary with firing rate on the training images. (a) Target-class top-1 accuracy drop after removing one concept for each class–concept entry, binned by firing rate. Teal shows the mean, orange shows the median, and counts appear above. The region below 0.2 sits outside B. (b) Top-1 accuracy drop after jointly removing every concept with firing rate below x. Gray shows random concept sets with the same activation mass, for thresholds where a mass-matched set could be sampled.

E SCHEDULE SENSITIVITY

The schedule of K decides whether each invariant is recovered and at which level. This appendix compares four schedules on B and checks the cluster-count rule on other matrices.

We ran four schedules on the training-derived matrix, which holds 1,227 shared concepts, with the clustering step and every other constant unchanged. They differ only in where the levels sit.

- The fixed uniform grid holds K to a preset ladder from 500 down to 56 in steps of 12, then 50, for 39 levels. It is the only schedule that ignores the classes that remain.
- The achieved-size schedule targets average cluster sizes on a preset linear grid from 2 to 20 and sets K from the remaining classes as in Section 2.2.
- The histogram sub-level schedule anchors levels at the FSS values that occur in the matrix and inserts extra levels between anchors, more where more concepts sit.
- The executed rule uses one level per occurring FSS value (Section 2.2) and nothing else, run through its terminal $K = 1$ level.

A fifth schedule places its levels at the peel sizes measured in the histogram sub-level run. It needs at least 20 measured peels per FSS value and cannot be calibrated on this matrix, which holds 11 concepts at FSS 6. On the validation-derived matrix it peeled 1,536 of 1,541 shared concepts, against 1,538 to 1,540 for the other adaptive schedules there.

Figure 8 shows the per-level peel record of each schedule. The fixed grid peels 1,183 of the 1,227 shared concepts and none of the eight concepts with FSS of 9 or more. Its peels concentrate at the finest levels and rise again in a mid-grid bump. The achieved-size and histogram schedules peel 1,225 and 1,227, and the executed rule peels 1,224 before its terminal levels. The spread among the three is 3 concepts, or 0.24% of the shared inventory. The terminal levels at $K = 2$ and $K = 1$ then peel the executed rule's last three. Schedule choice therefore moves concepts across levels but barely changes what is recovered, and only the fixed grid loses the broadest concepts.

The cluster-count rule across builds. The anchors s_l are read from each matrix's own FSS histogram, so the rule introduces no tuned parameter. On B the executed K_l equals the rule at every level (Appendix B). We checked three further matrices, each built from 50 images per class. One uses the validation images, one uses the validation images with the second autoencoder seed, and one uses a disjoint training sample. Across their 36 pre-exhaustion levels the executed K_l equals $\lfloor a_l/s_l \rceil$ under round half to even. Three levels are exact 0.5 ties for which round half up would give a different value. The occurring FSS values differ between builds, and each build's executed anchors match its own histogram. The Lasso penalty was chosen once on earlier matrices and was not tuned per build.

Stability of level placement. Before its terminal levels, the executed rule peels 1,224 shared concepts on B. The median ratio between the target size of the level where a concept peels and the concept's own FSS is 1.33. Across the two disjoint halves of the training images, concepts peeled in both receive the same level label at a rate of 0.512, while the peeled set itself reproduces at Jaccard 0.84. We therefore report the schedule and its per-level clusters, never a stable depth for any single concept.

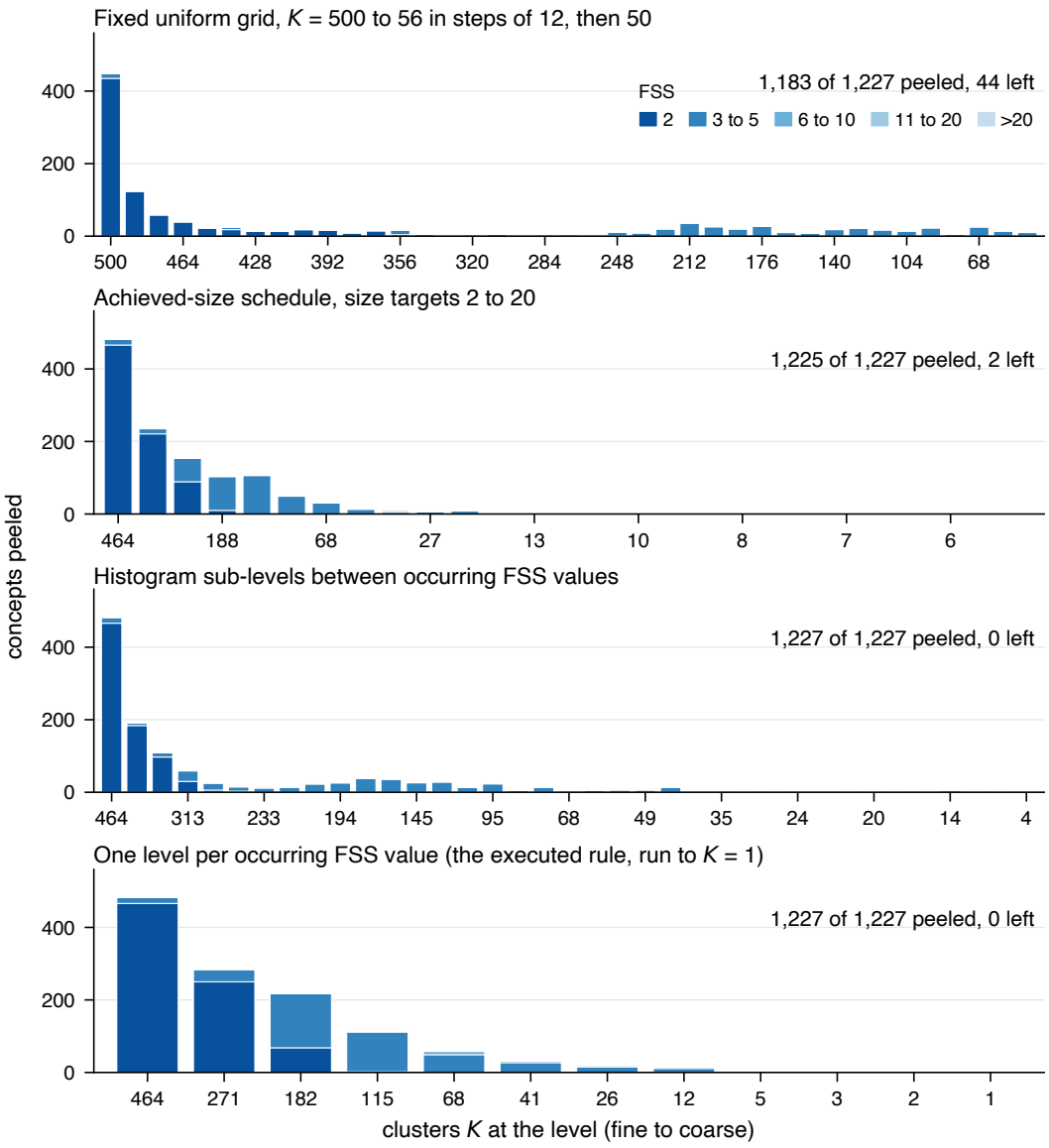


Figure 8: Concepts peeled at each level under four schedules on the training-derived matrix, stacked by the concept's FSS. Each panel is one schedule and runs fine to coarse from left to right. The bottom labels give the cluster count K at each level, and the text in each panel counts peeled concepts of the 1,227 shared concepts. The fixed uniform grid leaves 44 concepts unpeeled, including all eight with FSS of 9 or more. The executed rule is shown through its terminal $K = 1$ level.

F Nesting tests and the containment forest

Direct nesting test. A cluster can split at the next level and still find a parent later. In a single taxonomy, its remaining classes would stay together at every coarser level. For each cluster with at least two classes still active at the next level, the test asks whether those classes remain in one cluster there. Of the 885 such clusters on the training-derived schedule, 506 split at the next level (57.2%, or 57.5% at $K \geq 3$). The split fraction is 74.7% at the finest adjacent pair and at least 30% for 8 of the 11 adjacent pairs (Figure 9). On the validation-derived matrix, 809 of 1,229 split (65.8%).

Synthetic nested-tree control. Each level reclusters a residual matrix, so a split can come from the procedure rather than from the representation. We measure the procedure's share with synthetic

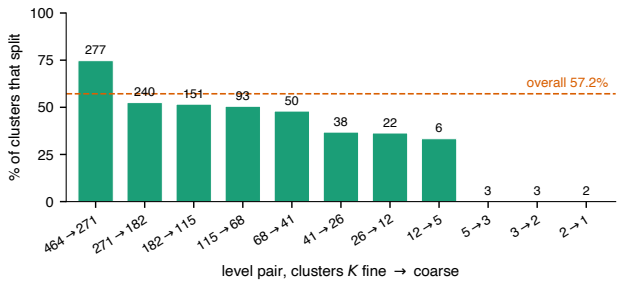


Figure 9: Direct nesting test. For each adjacent level pair, the bar gives the fraction of clusters with at least two classes still active at the coarser level whose surviving classes split across two or more clusters there. Counts above give the clusters tested. The dashed line is the overall 57.2% (506 of 885). The same test reports 20.0 to 36.8% splits on exactly nested synthetic matrices.

matrices whose true structure is a nested tree. The synthetic matrix keeps the 3,052 class unique columns of B. It replaces the 1,227 shared concepts with leaf sets of nodes in a random nested tree over the 929 classes that carry a shared concept. The bottom nodes form a random partition into sets of 2 or 3 classes and are always included. Every class is therefore covered, and the first level has $K_1 = 464$ as in the observed matrix. For each sharing count s the remaining columns draw from internal tree nodes of size s, with replacement when fewer exist. The matrices contain 489 to 498 unique supports among 1,227 columns, against 1,019 in B. For noise level ϵ, each membership moves with probability ϵ to a random class outside the concept's support. Every synthetic matrix runs through the same rule for K_l (re-derived from its own FSS values) and the same test. Three tree seeds give split fractions at $\epsilon = 0$ of 36.8%, 20.0% and 22.0%. At $\epsilon = 0.2$ they are 55.8%, 51.9% and 56.2%. Peeling assigns at least 1,223 of 1,227 concepts in every run. The observed 57.2% sits well above the exactly nested range. It also sits above the range at the noise level measured in B, given next, so the procedure accounts for only part of the observed splitting.

An earlier construction without guaranteed class coverage included 804 of the 929 classes. Its preregistered prediction was a split fraction below 10% at $\epsilon = 0$, and the observed value was 31.8%. The miss showed the procedure splits nested inputs far more than we expected, and the full-coverage construction above was built before we relied on the range.

At $\epsilon = 0$, K_l exceeds the number of distinct residual row patterns at most levels, and 64 to 74% of splits separate classes with identical rows. In the observed matrix, K_l stays below the distinct-row count at every level above $K = 1$, and 6 of the 506 splits separate identical rows. The synthetic splits therefore quantify what the procedure itself introduces.

Noise level read from the data. The 20% level above was set before the link noise was measured. We measured it in two ways. Rebuilding B from two disjoint halves of the training images keeps 92.6% of the shared links of one half in the other. A counting model of the link rule at each class's full image count predicts that 3.3% of the shared links drop. We therefore reran the synthetic trees at a noise level of 3.5% and at a stress level of 10%. The noise model moves each membership to a random outside class, so it adds as many spurious links as it drops, which favors the tree alternative. The synthetic runs stop below $K = 3$, so the observed reference at $K \geq 3$ is 506 of 880 clusters (57.5%). At 3.5% the three tree seeds split 22.4%, 25.7% and 18.8%. At 10% they split 37.7%, 36.2% and 40.2%. On the validation-derived matrix the same model gives a 12% level, at which the seeds split 46.4%, 45.6% and 49.2% against the observed 65.8%. At the measured noise level the identical-row share of synthetic splits is 6 to 26%, so these splits come from the noise rather than from a forced K. Three preregistered bands were missed on the low side. We predicted 22 to 45% at 3.5% noise and one seed fell to 18.8%. We predicted 20 to 45% at zero noise on the validation-derived matrix and two seeds fell to 16%. We predicted that duplicated supports would still make up over 35% of the synthetic splits at 3.5% noise and they made up 6 to 26%. None of the misses favors the tree alternative. A nested tree carrying the measured link noise reaches 22% on the training-derived matrix against the observed 57.5%.

The containment forest. The containment structure is derived after clustering and is not an output of the clustering step. Each cluster is a node. We remove the classes that have already been peeled away and search coarser levels for the first cluster containing all remaining classes. When at least two classes remain, an edge connects the node to this parent. A node whose classes are all peeled before any coarser cluster contains them is a root. No synthetic super-root is added, and the acyclic components form the containment forest of Section 3.1. The 71 classes that carry no shared concept never enter a cluster and count as single-node trees outside the 1,039 clusters below. As one concrete path, eight spaniel and setter classes occupy two clusters at the level with $s_l = 9$ and reunite in one cluster at the next level with $s_{l+1} = 10$. A stricter original-scope variant requires every original class to remain present.

The training-derived construction uses the levels from $K_1 = 464$ to $K_{12} = 1$ and excludes the pre-peeling partition of 1,000 single classes. Among its 1,039 nonterminal clusters with at least two classes, 36.5% find an immediate parent, 32.7% a skip-level parent and 30.8% are roots (Figure 10). Skip edges span 2 levels for 140 clusters, 3 for 102, 4 for 57 and 5 or more for 41. In all, 700 clusters (67.4%) find a parent at $K > 3$. Under original scope the fractions are 30.6%, 13.6% and 55.8%. At each parent level, the largest parent's share of retained child mass stays between 0.03 and 0.21 for $K \geq 12$. It is 0.39 at $K = 5$, 0.486 at $K = 3$, 0.872 at $K = 2$ and 1.0 at $K = 1$. The validation-derived schedule before its terminal extension has 1,346 clusters with fractions 31.2%, 38.3% and 29.6%. A further 0.9% keep two classes to the end and are roots as well. Its median skip is 3 levels and its largest parent share peaks at 0.557 at $K = 5$.

Visual and label similarity within trees. A tree's classes are the union of its clusters' classes, and 206 trees hold three or more. For each tree we take the mean pairwise similarity of its classes. The reference is 1,000 random sets of the same number drawn from the 929 classes that carry a shared concept. Three similarities are used, none of which depends on DINOv2 or on the autoencoder. The first is the cosine between centered class means of a supervised ImageNet classifier, ViT-B/16 (Dosovitskiy et al., 2021), over 50 validation images per class. The second is the cosine between standardized class means of 26 color and texture statistics on the same images. These are hue, saturation and brightness histograms with eight bins each, plus the mean and standard deviation of edge strength. The third is Wu–Palmer similarity. The trees exceed the random median in 99.0%, 98.1% and 93.2% of cases respectively. At every one of the nine qualifying levels, the within-cluster mean exceeds the shuffled 95th percentile on all three. We predicted 55 to 70% for the color and texture arm and observed 98.1%, so low-level appearance separates the trees far more than expected. The result is an association. A class's photographic setting varies with its label family, and the two visual arms cannot be separated here from the semantic one.

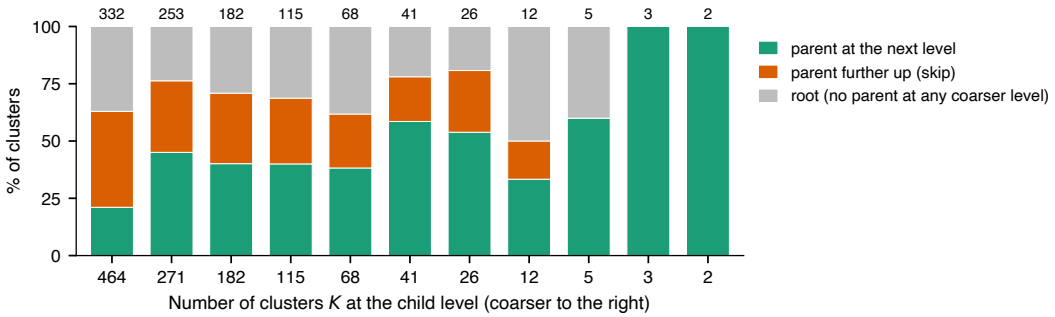


Figure 10: Containment outcomes for the training-derived schedule. Each bar is one peeling level, labeled by its number of clusters K. For clusters of at least two classes, teal marks an immediate parent, orange a parent at a later level and gray a root with no parent at any coarser level. Counts above give the clusters evaluated.

G WHAT ONE TREE HOLDS

A tree holds a shared concept at dilution d when its smallest node containing all the concept's linked classes has at most d times that many classes. MCP's dilution for a concept is the size of the cluster it

is peeled from divided by the number of classes it links to. Among pairs of shared concepts whose class sets overlap, 24.1% are nested and the rest cross, so no tree holds every concept at dilution 1. Table 7 gives the fraction of the 1,227 shared concepts held at each dilution by three arms. The arms are one average-linkage Jaccard tree on B, one random binary tree as the chance floor, and MCP. At dilution 2 the tree on B holds 762 concepts and leaves 465 unheld. MCP holds 891, of which 287 the tree on B leaves out, and 178 concepts are held by neither. We predicted that MCP would hold at least 90% at dilution 2 and it holds 72.6%. So MCP peels every shared concept, but 27% of them from a cluster more than twice their size. We also predicted that the concepts one tree leaves out would be broader than the held ones. They have the same median size of 2. The medians hide a smaller shift. Of the unheld concepts, 43% link three or more classes, against 32% of the held ones. Whether the unheld concepts are near-threshold links or crossing structure is open. The forest holds 129 more shared concepts at dilution 2 than the single tree on the same matrix. At dilution 1 the tree holds more, 44.7% against 31.7%.

Table 7: Share of the 1,227 shared concepts held within dilution d by one tree or by MCP's forest, in %. A concept is held at dilution d when the smallest tree node, or the cluster it is peeled from, contains all its classes and at most d times as many classes in all.

	$d \leq 1$	$d \leq 1.5$	$d \leq 2$	$d \leq 3$
One tree on B (Jaccard, average linkage)	44.7	56.1	62.1	68.5
One random binary tree	0.0	0.0	0.2	0.2
MCP, cluster at peel	31.7	58.4	72.6	88.8

H SUBSPACE SILHOUETTE ACROSS LEVELS

Figure 11 shows the statistical comparison of Section 3.2 level by level. This appendix defines the distance behind it. At a given level, let x_c be row c of B restricted to the concepts not yet peeled, scaled to unit length. For every cluster G we fit a basis U_G from the top right singular vectors of its rows. We keep 3 vectors, or the rank of the rows when that is smaller. The residual of class c against cluster G is

$$\rho_G(x_c) = x_c - U_G U_G^\top x_c. \tag{5}$$

Writing G_c for the cluster holding class c, the distance between classes c and c' is

$$\begin{aligned} D(c,c')^2 = \frac{1}{4}\Big| \, &\|\rho_{G_c}(x_c)\|^2 + \|\rho_{G_{c'}}(x_{c'})\|^2 + \|\rho_{G_{c'}}(x_c)\|^2 + \|\rho_{G_c}(x_{c'})\|^2 \\ &- 2|\langle \rho_{G_c}(x_c),\, x_{c'}\rangle| - 2|\langle \rho_{G_{c'}}(x_{c'}),\, x_c\rangle| \, \Big|, \end{aligned} \tag{6}$$

following the authors' released code (Lipor & Balzano, 2020). The outer absolute value is part of that code and never changes the distance, because the enclosed term cannot be negative. The silhouette is the standard one computed on D. The reference shuffles the cluster labels 10 times and refits every U_G per shuffle. The observed silhouette exceeds the shuffled reference at all eleven qualifying levels. Each has at least ten classes in total and at least two clusters, including one with multiple classes. The margin is smallest at the finest level (0.062) and widens toward coarser levels. Late levels hold few classes and their reference is noisy. The level with five clusters dips to a silhouette of 0.368, still about three shuffle standard deviations above its reference. On the validation-derived matrix the finest-level margin is smaller at 0.028.

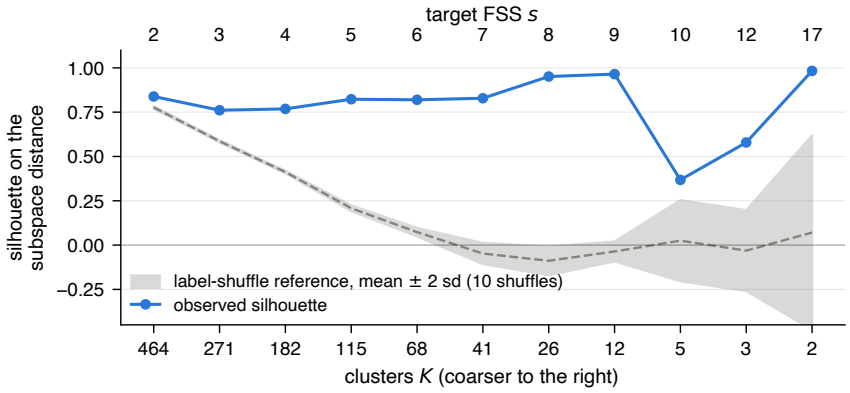


Figure 11: Silhouette on the Lipor–Balzano subspace distance at each level of the peel on the training-derived matrix. The shaded band is the label-shuffle reference, mean plus or minus two standard deviations over 10 shuffles. The observed value lies above the reference at all eleven qualifying levels.

I WordNet random-cluster null per level

Table 8 gives the per-level results behind the semantic comparison of Section 3.2. Every level with at least five clusters of two or more classes enters the comparison. At each such level, the observed mean medoid distance lies below the 5th percentile of the size-matched random null. The observed least common subsumer similarity lies above the null 95th percentile. At the finest level, one cluster holds exactly Irish Wolfhound, Otterhound and Scottish Deerhound. The final part of this appendix types the clusters that no single WordNet family covers.

The scores are defined as follows. Wu–Palmer similarity divides twice the depth of the least common subsumer of two classes by the sum of their depths (Wu & Palmer, 1994),

$$w(c, c') = \frac{2\,\mathrm{depth}\big(\mathrm{lcs}(c, c')\big)}{\mathrm{depth}(c) + \mathrm{depth}(c')}, \tag{7}$$

where $\mathrm{lcs}(c, c')$ is the deepest WordNet node that is an ancestor of both classes and depth counts steps from the WordNet root. For a cluster G, the medoid m is the class with the highest mean similarity to the rest. The medoid distance is the mean distance from the other classes to it.

$$m = \arg\max_{c \in G} \frac{1}{|G| - 1} \sum_{c' \in G,\, c' \neq c} w(c, c'), \qquad d_{\mathrm{med}}(G) = \frac{1}{|G| - 1} \sum_{c \in G,\, c \neq m} \big(1 - w(c, m)\big). \tag{8}$$

The subsumer similarity of G is the mean of $w\big(c, \mathrm{lcs}(G)\big)$ over its classes, with $\mathrm{lcs}(G)$ the deepest WordNet node that is an ancestor of every class in G.

Subsumer depth and medoid distance. Pooling the random clusters from the 25 draws at each level gives 26,000 clusters. We fit an ordinary least-squares line predicting medoid distance from the depth of the least common subsumer. Across the 1,040 observed clusters with two or more classes, depth and medoid distance have Spearman correlation -0.735. Of these clusters, 76.7% fall below the fitted null line. Part of this association is definitional, since deeper subsumers imply smaller Wu–Palmer distances and the two quantities are algebraically coupled for clusters with two classes. On the validation-derived matrix, size-matched random clusters alone give Spearman -0.60 against -0.77 for the observed clusters. The evidence is therefore the offset from the null. Observed clusters sit closer together than random clusters at the same WordNet depth.

Clusters outside WordNet families. Not every cluster falls within one WordNet family. Here we type each cluster by the largest share of it that a single family covers, and we call the remainder contextual. For each cluster with at least three classes, let f be the largest fraction of its classes covered by one WordNet family of depth at least 9. Depth 9 excludes umbrella nodes, since generic nodes such as artifact sit at depth 4. A cluster is full-family when $f = 1$, majority-only when

Table 8: Per-level WordNet comparison on the training-derived matrix. The columns report the number of clusters with at least two classes, the observed mean medoid distance, its null 5th percentile, the observed least-common-subsumer similarity, and its null 95th percentile. Null medians appear in parentheses.

s_l	clusters	medoid distance	null 5th pct. (median)	subsumer similarity	null 95th pct. (median)
2	332	0.281	0.533 (0.542)	0.794	0.589 (0.581)
3	253	0.295	0.515 (0.522)	0.750	0.565 (0.557)
4	182	0.287	0.497 (0.506)	0.737	0.537 (0.529)
5	115	0.282	0.480 (0.487)	0.712	0.516 (0.506)
6	68	0.267	0.460 (0.471)	0.719	0.523 (0.508)
7	41	0.269	0.455 (0.469)	0.718	0.518 (0.500)
8	26	0.315	0.470 (0.490)	0.691	0.493 (0.478)
9	12	0.305	0.448 (0.476)	0.566	0.458 (0.419)
10	5	0.313	0.410 (0.447)	0.596	0.528 (0.497)

$0.5 < f < 1$, and contextual when $f \leq 0.5$. The training-derived matrix has 632 clusters with at least three classes. Of these, 211 are full-family, 143 majority-only and 278 contextual. The validation-derived matrix has 871, split 195, 197 and 479.

Contextual does not mean non-taxonomic. WordNet's artifact branch is shallow. Of the 236 clusters whose common subsumer sits at depth 4 or less, 164 (69.5%) are contextual. We therefore never read the contextual count as a count of non-taxonomic relations.

Training-derived removals cover 205 contextual and 146 full-family clusters. The drop on the cluster's classes exceeds the outside drop for 202 of 205 contextual clusters (98.5%) and 141 of 146 full-family clusters (96.6%). Matched controls exist for 204 and 144 of them. The observed drop beats the mean control drop for 199 of 204 (97.5%) and 140 of 144 (97.2%). Median drops on the cluster's classes are 2.64 pp for contextual and 4.61 pp for full-family clusters. Majority-only clusters are excluded from this comparison.

J Substrate and dictionary-size comparisons

These comparisons are model-selection diagnostics, not controlled ablations of token substrate. The primary model is an 8,192-concept class-token-only autoencoder. Three 32,000-concept recipes on the same token leave 42%, 83% and 67% of concepts dead. The primary model leaves 11.5%, or 940 of 8,192.

A separate auxiliary analysis uses a count-based class–concept matrix built from 50 validation images per class. It links a class to a concept when the concept fires on at least 10 of the class's 50 images. Concepts with fewer than 10 fires over the full validation set are excluded first. This population differs from the 1,227 shared concepts of the primary rate-based matrix. An auxiliary peeled concept passes a coherence criterion when its supporting classes' count-matrix rows have mean pairwise cosine distance at most 0.5. The criterion is applied for cluster counts between 20 and 500. Three models are scored identically. The 8,192-concept model passes at 0.718 over 1,528 auxiliary peeled concepts. The earlier all-token reference passes at 0.636 over 1,071, and a 4,096-concept model passes at 0.487 over 1,867. The 4,096- and 8,192-concept models share the substrate, but the comparison does not isolate which consequence of dictionary capacity explains the difference.

In the auxiliary count-based matrices, the all-token reference has 34 concepts linked to more than 20 classes and the 8,192-concept model has two. This documents a breadth limit of the selected model. The all-token reference was trained earlier with a different recipe, so substrate and training procedure are confounded here.

K Limitations in full and release details

Limitations in full. Every result concerns the final-layer class token, one autoencoder training recipe and ImageNet-1k. The clustering, forest and coherence results come from one DINOv2 backbone. The peeling coverage and the class-level and cluster-level ablations are replicated on

DINOv1 and DINOv3 under the same recipe (Appendix C.2). On DINOv2, the second autoencoder seed covers peeling coverage, the cluster-count rule and the class-level ablation. It also covers the class group comparisons of Section 4.2. On DINOv1 and DINOv3, second seeds cover the class-level and cluster-level ablations (Table 5). The clustering, forest and coherence results are reported for one seed. The selected class-token model keeps few of the broadest shared concepts. In the auxiliary matrices of Appendix J, an earlier all-token autoencoder holds 34 concepts linked to more than 20 classes. The selected model holds 2, and the two recipes also differ. A class-token autoencoder gives one activation per concept per image, so patch-level heatmaps of a concept require a separate attribution method. A concept's peeling level is unstable across disjoint image halves, at 0.51 against 0.84 for the peeled set, so we report per-level clusters and never a per-concept depth. Breadth and high-FSS tail counts depend on the 14.0% boundary-flagged links, and we release the flags with B (Appendix D). Held-out images cover class-level removals only. The synthetic-tree range of Appendix F comes from one family of random trees and is not a calibrated threshold for hierarchy.

Release details. Beyond the matrix, peel records and per-level clusters of Section 3.1, the release includes the boundary flags and the synthetic matrices. It adds the result files behind every table, the seeded scripts and a server-free `index_standalone.html` copy of the explorer in the archive. The interactive explorer runs at `https://local-structure-explorer.vercel.app`. An archived copy is available at `https://anonymous.4open.science/r/local-structure-explorer-EEDF`.