CHARACTERIZING WEIGHT-LEVEL CONCEPT REPRESENTATIONS IN VISION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept-based explanations reveal which human-understandable concepts a network encodes, but mostly locate them in activation space, as directions or probes, rather than in the parameters that compute them. Mechanistic interpretability computation in weights and circuits, but typically for behaviors the model was trained to produce, such as a task, subtask or output class. We bring the two together: starting from human-defined concepts the model was never trained to predict, we learn sparse binary masks over the weights of frozen vision models that isolate the parameters needed to detect each concept. We first show that these masks are valid concept detectors, and then test whether this parameter-level structure is unique. Our main finding is that it is not: masks trained with different random seeds are equally valid but select different weights. The weights they all share often fail to detect the concept on their own, and ablating each mask changes the network's class predictions differently. A single mask is thus enough to detect a concept, but not to claim which weights compute it; such claims require several independently trained masks and a direct test of what they share.

1 INTRODUCTION

Image classification models have increasingly been deployed in domains where their behavior must be transparent, auditable, and explainable (European Parliament and Council of the European Union, 2025; Buolamwini & Gebru, 2018; Hansson et al., 2021). Understanding how their learned internal structures are connected to human-understandable concepts can help with these requirements since it explains their behavior and can trace which parameter-level structures support them.

Prior work approaches this challenge from two directions. *Mechanistic interpretability* (Olah et al., 2017; 2020; Cammarata et al., 2020; Elhage et al., 2022) and *neural network modularization* (Csordás et al., 2021; Lepori et al., 2023) start from the model. The former interprets the computation of given components, e. g., , neurons, channels or layers, while the latter searches among the weights for the parameters that implement a given sub-task. *Concept-based* methods (Poeta et al., 2023) instead start from human-defined concepts and locate or embed them in the network's activation space, e. g., , as directions (Kim et al., 2018, TCAV) or through a dedicated bottleneck layer (Koh et al., 2020, CBM;). Whether such concepts can also guide the discovery of parameter-level structures, as the model-first approaches do, and what those structures look like, remains largely unexplored. Answering this would ground concept-based explanations in the network's parameters rather than only its activations.

In this work we ask whether human-defined visual concepts can be localized and detected based on a sparse parameter-level structures in a trained visual network, even when the concept was not an explicit training objective. We start with semantically meaningful attributes, e. g., shape and color. We define concept as annotations along a dataset. This used to train a network to perform a visual classification task, with the network not being trained to explicitly recognize the concepts. We then use an adapted Continuous Sparsification (CS) (Savarese et al., 2020), adapted to a concept-conditional objective in place of its task-level one. We use this to search for concept-specific sparse parameter masks over the frozen network. CS is a pruning technique related to the Lottery Ticket Hypothesis (LTH) shown to discover sparse subnetworks that support learned behavior, making it a natural experimental technique for searching for concept-specific parameter structures.

However, following LTH identifying a sparse parameter-level representation raises a further question than just if it is valid or not: is the discovered structure a unique representation of the concept, or one

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.

Table 1: Comparison of XAI methods across key dimensions. **Explanation unit**: the model component or space in which the explanation is expressed. **Multi-layer**: traces concepts across multiple layers. **Model-level**: produces explanations independent of a specific input. **Multiplicity**: assesses whether repeated runs of the method recover the same explanation.

Category	Method	Explanation unit	Concept definition	Multi-layer	Model-level	Multiplicity
Feature attribution	LIME Ribeiro et al. (2016)	Input	—	✗	✗	✗
	SHAP Lundberg & Lee (2017)	Input	—	✗	✗	✗
Concept attribution	TCAV Kim et al. (2018)	Activation	User-specified	✗	✓	partial
	CRP Achtibat et al. (2023)	Act. / Input	User-specified	✓	partial	✗
Concept circuit discovery	ADVC Rajaram et al. (2024)	Neuron	User-specified	✓	partial	✗
	VCC Kowal et al. (2024)	Activation	Unsupervised	✓	✓	partial
	GCC Kwon et al. (2025)	Neuron	Query-driven	✓	✗	✗
	Ours	**Weight**	User-specified	✓	✓	✓

of several possible realizations? We find that shuffling the order in which the data is presented to the our CS adaption, across independent runs, leads to different masks that exhibit comparable concept-specific behavior. This observation is in line with the LTH, that there might not be a unique set of parameters that represent a concept or function but rather a multitude of concept-specific parameters that form functionally almost equivalent masks. The aim of this study is to better understand and analyze these concept masks. To characterize and better understand the multitude of concept-specific masks we train three CNN-based architectures and one VIT on two concept annotated datasets. Based on those backbone models we use random initializations to construct a set of concept-specific masks across several concepts. Following this we evaluate if the masks are valid concept detectors at all by checking if the concept-masks can discriminate between concept positive inputs and concept negative inputs. Once validity is established we carefully study the multitude of concept-specific masks along their *structure*, *functionally* and *behaviorally*. We find that each mask acts as a valid concept detector, performing comparatively with a linear probe such as CAV. Furthermore, we find that there is not a single unique mask that represents a concept. When running 30 seeds we find measurably different masks. While, the masks agree almost perfectly in *function* on whether a concept is present or not, they shift the networks *behavior* when applied. The main takeaway is that a single mask is enough to detect a concept but any mechanistic claim about which weights implement it and interventions built on those weights, should rest on several independently trained masks rather than one.

2 Related Work

Concept attribution techniques such as TCAV Kim et al. (2018) and CRP Achtibat et al. (2023) rely on activation-space directions and concept-conditioned relevance propagation, respectively. They explore activation patterns related to a concept without identifying the related weights. A complementary field moves beyond attribution to trace how visual concepts are implemented across multiple layers as computational subgraphs. Rajaram et al. (2024) introduce Automatic Discovery of Visual Circuits (ADVC) as a cross-layer attribution to iteratively identify neuron-level circuits from a small set of concept examples, producing a single unified circuit per concept. ITCAV (Kowal et al., 2024) discover human-understandable concepts and their inter-layer connectivity in a fully unsupervised manner, representing concepts as cluster centroids in each layer's activation space and quantifying their cross-layer relationships. In terms of concept subnetwork, Granular Concept Circuits (GCC) (Kwon et al., 2025) is closely related to our work. It builds per-query concept circuits via directed acyclic graphs over neurons, selecting connections via Neuron Sensitivity Scores and Semantic Flow Scores to ensure both functional dependency and semantic alignment between connected neurons. Despite their differences, none of them addresses which weights in a pretrained network are responsible for encoding a predefined concept. Table 1 contrasts these approaches.

Concept subnetwork discovery Early work in mechanistic interpretability aimed at discovering which concept a single neuron encodes Bau et al. (2017; 2020). These approaches look neurons in isolation, disregarding the possibility that concepts can be encoded across many units and layers. Olah et al. (2020) addressed this with the circuits framework, showing that through manual analysis that networks implement human-interpretable features as sparse subgraphs of connected neurons. The Lottery Ticket Hypothesis (Frankle & Carbin, 2019), in parallel, demonstrated that every network contains a sparse subnetwork that can match the full model's performance. Csordás et al. (2021) rely

on this subnetwork search and learn weight masks on frozen networks to probe functional modularity, finding task-specific subnetworks that are often not reused across tasks. Lepori et al. (2023) extended this to compositional tasks, using CS (Savarese et al., 2020), to extract subnetworks that implement individual *subroutines*. For Vision Transformers, recent work recovers class-specific, edge-level circuits via activation patching (Zukowska et al., 2026), while SAEs (Stevens et al., 2025) decompose the residual stream into interpretable features and cross-layer transcoders (Chatzoudis et al., 2026) trace their interactions across layers.

However, mask-based subnetwork discovery (Csordás et al., 2021; Lepori et al., 2023) has been validated on tasks where recovered subnetworks can be checked against ground-truth subtasks; for concepts in standard vision models, no such reference exists. Besides, while recent work shows that attribution-based circuits vary across data shuffles (Frankle et al., 2020; Méloux et al., 2025), it remains unknown whether independent runs of mask-based discovery recover the same subnetwork. Without this check, discovered subnetworks cannot be distinguished from many equally valid solutions. We address both gaps by studying the run-to-run reproducibility of concept subnetworks in CNNs and ViTs.

3 METHOD

Our starting point is a network trained to perform a classification task and a set of semantically meaningful concepts, unseen during training, and apply CS (Savarese et al., 2020) to learn sparse binary weight masks over the frozen backbone. For each concept, a binary mask is optimized to maintain the dense model's internal representation on concept-present samples and to suppress the concept's associated classes when the concept is absent, while being penalized for keeping weights. The recovered masks also serve as annotation-free concept detectors at test time, requiring no probes, segmentation masks, or additional labeled data.

3.1 CONTINUOUS SPARSIFICATION FOR CONCEPT MASKS

Let $f(\cdot;\, w)$ be a vision-based Deep Neural Network (DNN) trained to map input samples $x \in \mathcal{X}$ to labels $y \in \mathcal{Y}$ using a set of weight parameters $w \in \mathbb{R}^d$. After freezing w, we introduce a set of human-specified binary concepts $\mathcal{C}$. Following Poeta et al. (2023), we call concepts high-level and semantically-meaningful attributes of the task at hand, such as *shape round* for traffic sign recognition or *furry* for species identification. For each concept $c \in \mathcal{C}$, we denote by $c(x) \in \{0, 1\}$ whether c is present in sample x. Let also $\mathcal{D}_c = \{(x_i,\, c_i,\, y_i)\}_{i=1}^n$, be a concept-labeled dataset partitioned into positive and negative subsets $\mathcal{D}_c^+ = \{(x_i, c_i, y_i) \in \mathcal{D}_c : c_i = 1\}$, and $\mathcal{D}_c^- = \{(x_i, c_i, y_i) \in \mathcal{D}_c : c_i = 0\}$.

For a fixed concept c, we seek a binary mask $m^{(c)} \in \{0, 1\}^d$ that identifies the minimal set of weights the frozen network uses when processing concept-present inputs. The masked model is

$$f_c(x;\, w, m^{(c)}) = f\Big(x;\, w \odot m^{(c)}\Big)\,,$$

where w remains fixed and only $m^{(c)}$ is optimized.

Relaxation and objective. Given that direct optimization over $\{0, 1\}^d$ is intractable, we follow Savarese et al. (2020) and learn a continuous relaxation $m^{(c)} \in [0, 1]^d$ defined by

$$m^{(c)} = \sigma\Big(\beta\, \theta^{(c)} + \delta\Big)\,,$$

where $\theta^{(c)} \in \mathbb{R}^d$ are trainable logits, σ is the sigmoid function, and $\beta > 0$ is a temperature factor to be annealed. To ensure stable gradient flow early in optimization, we initialize the values of $\theta^{(c)}$ as 0. Also, we anneal the temperature β upward from β_{init} to $\beta_{\text{final}} \gg \beta_{\text{init}}$, driving the mask toward binary values as optimization converges. The scalar offset $\delta > 0$ is our addition to the original formulation. It prevents the mask to halve the weights values at initialization, allowing the optimization to focus on weight selection instead of recovering performance in early stages.

Changes to the original formulation happen also in the loss function. Savarese et al. (2020) optimize masks for sparsity while maintaining global task performance. Our goal, on the other hand, is to discover sparse subnetworks that are sufficient and specific for a given concept c. Therefore, our

training focuses on reproducing the dense model's internal representation on the positive subset, and penalizing the mask when negative samples behave as positive. We optimize:

$$\mathcal{L}(c) \;=\; \mathcal{L}_{\text{suff}}(c) \;+\; \gamma(c)\,\mathcal{L}_{\text{spec}}(c) \;+\; \lambda \left\| m^{(c)} \right\|_1 \tag{1}$$

where the ℓ_1 is the sparsity term from CS and $\mathcal{L}_{\text{suff}}(c)$ and $\mathcal{L}_{\text{spec}}(c)$ are the sufficiency and specificity terms defined as follows:

$$\mathcal{L}_{\text{suff}}(c) = \mathbb{E}_{(x,c,y)\,\in\,\mathcal{D}_c^+}\left[\left\|\phi(x;\, w \odot m^{(c)}) - \phi(x;\, w)\right\|_2^2\right] \tag{2}$$

$$\mathcal{L}_{\text{spec}}(c) = \mathbb{E}_{(x,c,y)\,\in\,\mathcal{D}_c^-}\left[\sum_{k\in\mathcal{Y}_c} p_\theta(y = k \mid x)\right] \tag{3}$$

$\mathcal{L}_{\text{suff}}(c)$ measures the mean squared error between ϕ_{sparse} and ϕ_{dense} over the positive set, with $\phi(x; w) \in \mathbb{R}^p$ denoting the pre-classifier feature representation of f; i. e., the network's activations immediately preceding the final classification layer. Minimizing this distance equals to selecting the weights that are sufficient for keeping the performance of the masked model on the positive set of a given concept.

$\mathcal{L}_{\text{spec}}$ uses the classifier's output logits on the negative set to compute the sum of the probability mass landing on the classes related to the positive set. This term penalizes the mask when it behaves as if it saw the concept where the concept is not, yielding a concept-specific mask. We scale γ per concept by a purity-based risk score to prevent collapse on high-confusion concepts (see Appendix A). Within each epoch, mini-batches are drawn with equal numbers of positives and negatives, by subsampling the larger side to the size of the smaller one, i. e., a fresh random subset each epoch. Once training is completed, the binary mask is recovered by thresholding: $m_i^{(c)} = \mathbf{1}[\theta_i^{(c)} \geq 0.0]$.

Concept detection. We detect a concept by passing the masked model's pre-classifier features through the frozen classification head, i. e., the original, unmodified output layer of f. The head is never retrained or adapted to the mask, so any change in its output reflects what the masked features do to the original decision boundary. For an input x, the detection score is the softmax probability mass that the frozen head places on the concept's associated classes $\mathcal{Y}_c$:

$$s_c(x) \;=\; \sum_{k\in\mathcal{Y}_c} p_\theta\Big(y = k \mid x;\, w \odot m^{(c)}\Big)\,. \tag{4}$$

We report the ROC AUC of s_c on the test split. The same score is used throughout, to allow direct comparison.

3.2 Datasets, architectures and cells

Datasets. To better assess how concepts map to parameter-level structures, we use two concept-annotated datasets based on the different properties they provide: German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2011) and Animals with Attributes 2 (AWA2) (Xian et al., 2019). Even though both have class-level concept annotations, GTSRB's 43 concepts (shape, color, digits and pictograms) follow deterministically from the traffic-sign specification. This means that concept presence is assigned by construction, requiring no judgement call and is correct at the level of the individual sign, not only as a class-level aggregate. Differently, AWA2's 85 attributes are annotated via crowd source as class-level judgments over 50 animal classes. So, while GTSRB represents a clean case, in which masking failures can be attributed to the method rather than to label noise, AWA2 is the messier case, closer to what can be found in practice, where most concepts are spread out across classes and several are not visually grounded, e. g., habitat or behavior. Characterizing both lets us see which of our findings hold only under clean, well-behaved concepts and which persist under noisier, more realistic ones.

Architectures. Our results mainly rely ResNet-18, DenseNet-121 and EfficientNetV2. They are standard convolutional backbones that differ in how layers are connected and scaled: residual connections,

dense concatenation of earlier features, and compound-scaled blocks with depthwise convolutions and squeeze-and-excitation, respectively. In our experiments, we mask only the convolutional layers of the CNNs. To assess whether our findings extend beyond convolutional inductive biases, we additionally include a Vision Transformer (ViT), which processes images as sequences of patch tokens via self-attention rather than convolutions. In our experiments, we mask only the convolutional layers of the CNNs and, for the ViT, the MLP and attention (Q, K, V, output) projection weights in each encoder block. Further information on mask implementation, as well as on hyperparameters and training regimes for both the frozen backbone and the masks can be found in the Appendix (C, D and E)

Concepts. We selected 10 concepts from GTSRB and 9 from AWA2 using two statistics computed from the annotations alone: the number of concept-present classes (N_{classes}) and its *purity*, i. e., the share of concept-positive samples that belong to the most frequent class. The list comprises concepts present in one to over 40 classes, and purity from 0.05 to 1 (See Appendix B).

Cells and seeds. Each dataset-architecture-concept combination is a *cell*, giving $3 \times (10 + 9) = 57$ cells. For each cell we train $N = 30$ masks $\{m^{(c,r)}\}_{r=1}^{N}$ that share the same training configuration and differ only in the seed r that shuffles the training data. Note that the configuration may therefore differ between cells. Appendix E lists the values. Results are computed on the standard held-out test split of each dataset.

3.3 VALIDITY

Candidate masks must first be evaluated on whether their selected weights actually represent the concept and are not some spurious correlation. We use five criteria.

Sufficiency is the detection AUC of the masked network, i. e., how well it separates concept-present from concept-absent samples. **Specificity** is the fraction of concept-absent test samples the masked network does *not* assign to any class in the positive set. In this criterion, a concept-absent sample counts as a false detection if the predicted (argmax) class lies in $\mathcal{Y}_c$, and specificity is one minus the fraction of such samples. **Sparsity** is the kept fraction $\|m^{(c)}\|_1/d$. These three criteria are explicitly enforced at training time via the three terms in the loss function. As a baseline for those criteria, we fit a CAV (Kim et al., 2018) per cell on frozen pre-classifier activations, and score with the same AUC.

The remaining two criteria are evaluated post hoc and are not trained for. **Necessity** asks whether the mask's weights are needed for detection. We ablate them, i. e., apply the inverse mask $\mathbf{1} - m^{(c)}$, and score the concept with the concept detector. A necessary mask drives this AUC to chance (≈ 0.5). We report the AUC of the kept and the inverse mask side by side. **Redundancy** asks whether the concept is still *recoverable* from the ablated network. We refit a linear probe on the ablated features and define

$$\text{redundancy} = \text{AUC}_{\text{refit}}(\text{ablated}) - \text{AUC}_{\text{frozen head}}(\text{ablated}).$$

A large gap means the concept information survives ablation in the remaining weights but in a direction the dense model no longer uses. In that case, necessity of the readout does not imply that the information itself is confined to the mask. We report the same gap on the intact (kept) features as a reference, where it should be near zero.

Nulls. To understand whether the AUCs are actually informative, we compare against random masks with the same retention rate. Large-magnitude weights matter more to a network whether or not they encode the concept, so we use magnitude-matching nulls. For each discovered mask (one per seed) we draw, per layer, the same number of weights at random with probability proportional to their absolute magnitude in the dense network. For necessity, each null mask is inverted in the same way as the real one and scored with the frozen detector. As additional controls for necessity, we score uniform random masks at the kept density and at the complement density. Because 30 draws give a wide null range and a test would have almost no power, we report null ranges and no p-values (Section 4). Specificity and sparsity have no null.

3.4 MULTIPLICITY

Simply adding SGD noise, i. e., shuffling the dataset, can lead the optimization to different solutions (Frankle et al., 2020). Given the task of identifying the weights that compute a concept in a model, having several equally-performing masks raises the need to characterize the solution set. Masks are *equally performing* if their AUC is within ε of the cell's best, i. e., an ε-Rashomon set over masks (Fisher et al., 2019). In this work, we use $\varepsilon \in \{0.01, 0.02, 0.05\}$ and flag cells that lose seeds.

Structurally, we ask if the seeds keep the same weights. For each pair of masks $r \neq r'$, we compute Cohen's $\kappa = (p_o - p_e)/(1 - p_e)$ over the binary weight-inclusion vectors. p_o is the fraction of weights on which both masks agree (i. e., both keep or both drop) and $p_e = q_r q_{r'} + (1 - q_r)(1 - q_{r'})$ is the agreement expected from independent masks with densities $q_r, q_{r'}$. So $\kappa = 0$ is chance and $\kappa = 1$ is identical. Unlike Jaccard/IoU, κ corrects for density, making it comparable across cells with densities of different orders of magnitude (Appendix G). We also report the *core fraction* and *union inflation*: the sizes of the all-seed intersection and union relative to a single mask.

Functionally, we ask whether the weights that all seeds agree on are still able to detect the concept. Let $r \in \{1, \ldots, N\}$ index the $N = 30$ seeds of a cell in a fixed order, and let $m^{(c,r)}$ be the mask trained to detect concept c under the r-th seed's dataset shuffle. For $t \in \{1, \ldots, N\}$, let

$$m_t^{\cap} = \bigwedge_{r \leq t} m^{(c,r)}, \qquad m_t^{\cup} = \bigvee_{r \leq t} m^{(c,r)}$$

be the intersection and union of the first t masks under this ordering. By adding seeds one at a time, we measure the detection AUC of the running intersection, and call $m_N^{\cap}$ the *agreed core*. We compare it with the intersection of t random masks of matching density, and report the gap to this null and the gain of the union over the intersection.

Behaviorally, we ask if the masks make the same predictions. For a given input x, the N masks in a cell produce probability vectors $p_r(x)$ over K outcomes. Rashomon Capacity (RC) (Hsu & Calmon, 2022) measures how distinguishable these predictions are across masks. Formally, $\mathrm{RC}(x) = 2^{\max_\pi I(\pi;Y|x)}$, where π denotes the mask and Y its output distribution. RC is 1 when all masks produce the same prediction and approaches $\min(N, K)$ as their predictions become distinguishable. Unlike threshold-based measures, it uses the full probability vector and therefore captures differences in prediction confidence.

4 VALIDITY

In this section, we evaluate whether the sparse parameter-level masks are valid masks and actually connect to the human-understandable concepts they were trained for. Each panel in Figure 1 shows one of the criteria defined in Sect. 3.3. The three criteria in the upper half of the figure are compared against the corresponding null distributions obtained from randomly selected masks.

For **sufficiency**, top-left panel, we find that the large majority of the concept masks is well above the sparsity-matched randomly sampled null draws, with two exceptions that land inside their nulls observed range rather than above it. This strengthens our assumption that the retained weights are enough to detect the concept, and are better than a randomly drawn subset of weights. **Necessity**, the top-middle panel, shows the AUC of the inverted mask is compared against its sparsity-matched null distribution. For most cells, the real and null-mean AUC both sit within a few points of chance. Hence, ablating the concept mask and ablating a random mask of the same size are equally destructive. However, a handful of cells at the far end clear their null by a wide margin, with real AUC well above chance despite the ablation, and, for these, the necessity claim holds.

Redundancy, top right panel, shows on the x-axis that the gap between the frozen head and a new refit probe on the mask is close to zero in 54 of 57 cells. However, after ablation (y-axis), this difference is substantial (median 0.26, range 0.02–0.58). This means that the concept is still linearly decodable from the remaining weights, though not through the original classification head. Thus, removing the mask breaks the readout but does not remove the information. Some exceptions are: EfficientNetV2, whose ablated gap is small; and three AWA2 cells, to which a refit already helps on the mask, but adds nothing after ablation. Because we compare the ablated gap only to the kept gap and not to a null, we treat the gap as descriptive and draw no pass/fail conclusions. **Specificity**,

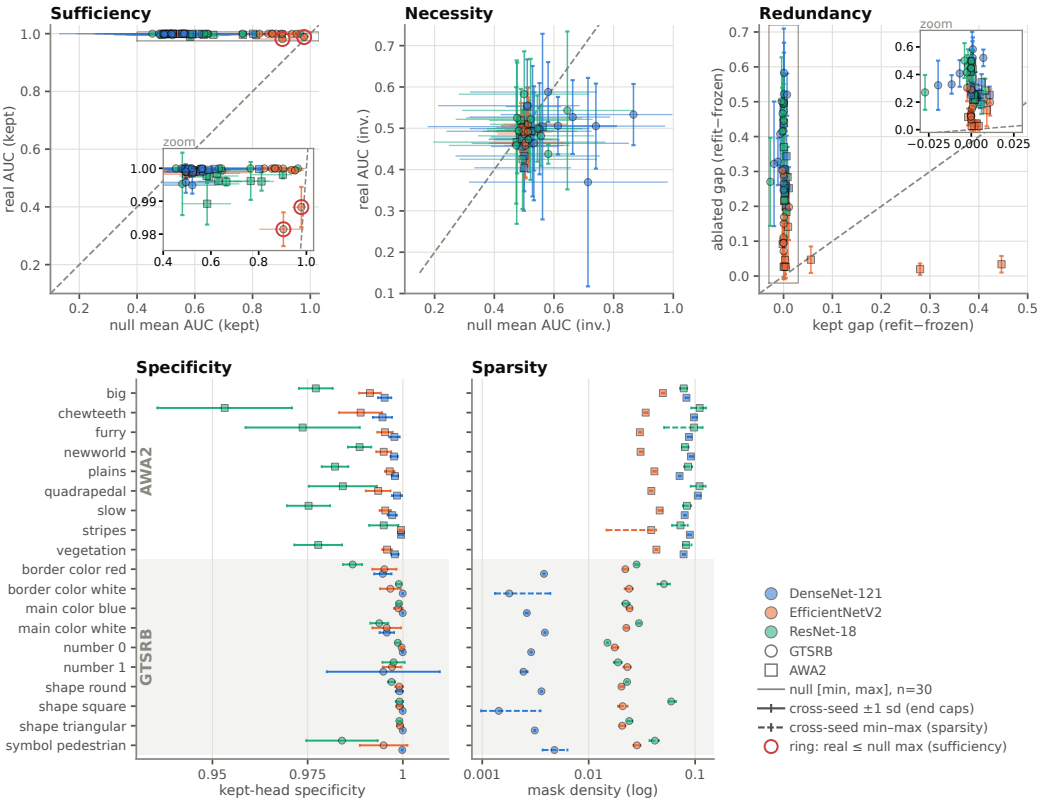


Figure 1: **Validity of concept masks across all 57 cells.** *Top:* real vs. null AUC; cells pass above the diagonal for sufficiency and redundancy and below it for necessity. *Bottom:* specificity and mask density, which have no null. Real scores show the cross-seed mean ±1 s.d. (unclamped, so whiskers can exceed 1.0); all other encodings are given in the legend.

lower-left plot on the figure, stays near 1.0 throughout (0.953–1.000) and is tightly clustered across all three architectures on GTSRB. The wider spread appears on AWA2, particularly for ResNet-18 on several concepts, which could be due to AWA2s more ambiguous and semantic concepts. **Sparsity** serves mainly to rule out that no architecture is systematically denser or sparser than the others across the grid, so EfficientNetV2's weaker necessity and redundancy cannot be explained by it simply using larger or smaller masks. Full results for all cells can be found in Appendix F.

Regarding ViTs (Appendix I), our initial analysis suggests that concepts represented in this architecture behave similarly to CNNs in terms of sufficiency, redundancy and specificity. Regarding necessity, the raw inverse AUC lies below the null mean or below the minimum in 10 out of 11 cells. On sparsity, however, ViTs keep more weights than CNNs on average.

Lastly, we compare our mask based concept detection against a flat CAV (logistic regression on frozen pre-classifier activations) (Kim et al., 2018) across the same 57 cells. Table 11 in Section F.1 reports the full comparison. On 38 of 57 cells, both methods perform near 1.0 and are statistically indistinguishable, with mean AUC difference below 0.001. Therefore, it is not possible to establish a general advantage on either side. Regardless, the strength of our approach, lies on identifying which weights carry the concept, grounding the concept representation in terms of the actual model's computation and cannot be achieved by simply probing the feature space.

5 Multiplicity

Section 4 showed that the sparse masks are valid concept detectors. In this section, we follow the lineage of Mechanistic Interpretability and investigate whether the different concept masks are

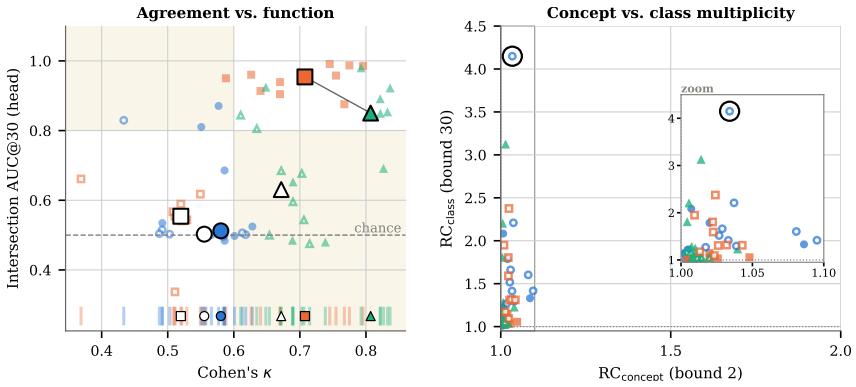


Figure 2: **Multiplicity summary.** Points are the 57 (dataset × architecture × concept) cells; circle/square/triangle = ResNet-18/EfficientNetV2/DenseNet-121, filled = GTSRB, open = AWA2. *Left*: mean pairwise Cohen's κ between the 30 seeds' masks vs. the detection AUC of the all-seed intersection at k=30. Large markers are group medians, and the rug along the bottom shows the per-cell κ values. The line joins the two GTSRB groups whose ordering inverts. *Right*: Rashomon capacity on the concept decision and on the class posterior. The circle highlights the extreme case .

the same solution or if they correspond to different solutions that happen to score alike. Figure 2 summarizes the results.

First, we evaluate the cell admissions at ε-Rashomon sets. At $\varepsilon = 0.01$, 48 of 57 cells comprise all 30 seeds, growing to 53 of 57 at $\varepsilon = 0.02$. At this ε, the four remaining cells consist of at least 28 seeds. At $\varepsilon = 0.05$ all cells comprise all seeds. We report these as flags rather than exclusions, and give admitted counts at $\varepsilon \in \{0.01, 0.02, 0.05\}$ for every cell in Appendix G.

Structural agreement. None of the cells have masks that completely agree with each other. With κ between 0.34 and 0.84 a typical pair of masks shares between a third and most of its weights, but never all of them. Similarly, in the intersection where 8% to 70% of a single masks weights are shared between seeds. While the union of all masks is 1.3 to 3.6 times larger than a single mask. The main observation is the difference stemming from dataset and architecture pairs with DenseNet-121 on GTSRB having the highest agreement (median $\kappa = 0.81$) and AWA2 with EfficientNetV2 the lowest (median $\kappa = 0.52$). The rug in the left panel of Figure 2 shows how the κ values are distributed across these pairs. Hence a single mask is one of a family of solutions whose spread depends on the backbone and data.

Looking at this Layer-wise, shows that weight sharing is not uniform across a networks depth. By looking at them pairwise, structural agreement between seeds is highest in early layers and falls as depth increases. Differently, concept selectivity, i. e., how well a layers masked weights detect their own concept against how well they'd respond to a different concept, is weakest early and strongest late. Layer-wise agreement and selectivity profiles, together with a null checks, are reported in Appendix H.

Functional agreement. Having a high structural agreement does not necessarily imply higher concept detection over the shared core. The left panel of Figure 2 suggests a positive correlation between κ and the concept detection AUC of the agreed core (Spearman's $\rho = +0.44$). However, this relation can be attributed more to the dataset-architecture group than to κ itself, since cells from the same group cluster together in κ and AUC and, and once each group's median is subtracted, the correlation drops to -0.05 (Appendix G.4). To aid the visualization, we make a purely descriptive quadrant split at $\kappa = 0.6$ and AUC $= 0.8$. The diagonal share highlights how uniform the groups are, since cells within the same group most often sit on the same quadrant. The main diagonal, with high κ and high AUC or low κ and low AUC is populated by the majority of cells (38/57). The remaining 19 cells fall off-diagonal. Fifteen cells (26%) have high κ but low AUC, showing that seeds can agree strongly on which weights to retain while the agreed core remains insufficient for concept detection. These cells are concentrated in DenseNet-121, including 7 AWA2 and 4 GTSRB cells. Only four cells have low κ but high AUC, indicating that strong concept detection can also arise despite limited weight agreement. The group-level results further illustrate the group effect.

On GTSRB, DenseNet-121 has higher weight agreement than EfficientNetV2 ($\kappa = 0.81$ vs. 0.71), but its agreed core has lower AUC (0.850 vs. 0.954). A similar inversion occurs on AWA2 between ResNet-18 and EfficientNetV2.

ResNet-18 provides an extreme case. The agreed core is small and performs near chance, with median AUC at k=30 of 0.512 on GTSRB and 0.503 on AWA2, but the corresponding union performs well at concept detection (AUC gains of 0.49 and 0.48 over the intersection; Appendix G.4). This indicates that, for this backbone, the concept is implemented by mostly disjoint masks, whose aggregation preserves concept-detection performance, whereas their shared core largely destroys it. Overall, 29 of 57 cells exceed the null AUC by more than 0.10; for GTSRB with EfficientNetV2, all ten cells do. More results can be found in Appendix G.4.

Behavioral agreement As described in Section 3, RC measures how different the outputs of the masks are by comparing the probability scores on each sample. The right panel of Figure 2 plots the RC at the class-level (RC_{class}) against the RC at the concept-level ($RC_{concept}$). Since we have 30 different seeds, RC_{class} has at most 30 different outcomes, while the binary concept detectors has $RC_{concept}$ at maximum 2. The concept-level capacity agrees substantially, never exceeding 1.095 in any of the 57 cells. Differently, on the class-level, the output of the masks within a cell can vary a lot with up to 4.15 (circled). Therefore, even though the masks can be almost interchangeably used to detect concepts, the same cannot be said in relation to class prediction. Threshold-based metrics and further analysis are reported in Appendix G.5.

On ViTs (see Appendix I), in 10/11 cells, all 10 masks are admitted at $\varepsilon = 0.01$. Structurally and functionally, their agreement is similar to representations on CNNs. Behaviorally, both RC_{class} and $RC_{concept}$ are lower, with RC_{class} about 1.0 in 8/11 cells.

6 Discussion and Conclusion

In this work, we explored how weight-level sparsification, specifically CS, can ground concept-based explanations in the parameters of DNNs. We trained binary masks over the weights of frozen backbones, each asked to reproduce the dense model's pre-classifier features on concept-positive inputs with as few weights as possible. Our analysis span 19 concepts on two datasets and four architectures (three CNNs and one VIT), repeating each cell 30 times with different data shuffles. Following the literature on concept circuit discovery, we first studied whether these masks are valid concept detectors, and then characterized how the masks found across the 30 seeds relate to each other. Key insighs we draw from or analysis are:

One mask is enough to detect a concept. We show that masks are sufficient, specific and sparse in nearly every cell and detect concepts as well as a CAV (Section 4). Additionally, masks from different seeds agree behaviorally at the concept level, by making nearly the same concept decision on the test set (Section 5).

No single set of weights computes the concept. Although many different masks are able to detect the concept, the same does not happen for the agreed core in most cells, even when structural agreement is high (Section 5). Together with the necessity floor (Section 4), this finding is consistent with concepts spread across weights that also serve other features (Elhage et al., 2022). Claims about which weights compute a concept therefore need an ensemble of masks and a direct test of their core. Besides, even when the core works, masks that agree on the concept can push the class prediction in different directions for the same inputs (Section 5). Hence, our analysis reaches two main conclusions. First, we show that human-defined concepts can be connected to parameter-level structures in the network. Second, we demonstrate that such structures are not unique, and treating one found mask as *the* concept representation is misleading.

Limitations. Our analysis is performed over datasets rather than single inputs, so a working core is evidence about how the model represents a concept, not about why one image received its label. We also do not test whether the dense model's decision passes through the masks: ablation is dominated by generic damage (Section 4), so this requires a targeted intervention study. Thirty seeds sample the family of valid masks but cannot bound its size. Furthermore, other sources of variation than the dataset shuffle may yield different solutions. Finally, our concepts are annotated at class level, so concept and class are partly entangled.

AI USE STATEMENT

We used LLM-based assistants to edit and shorten the text of this paper. All research ideas, experiments, analyses and conclusions are the authors' own, and the authors take full responsibility for the content.

REFERENCES

Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nat. Mac. Intell.*, 5(9):1006–1019, 2023. doi: 10.1038/S42256-023-00711-8.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3319–3327. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.354.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91. PMLR, 2018.

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 2020. doi: 10.23915/distill.00024.003. https://distill.pub/2020/circuits/curve-detectors.

Gerasimos Chatzoudis, Konstantinos D. Polyzos, Zhuowei Li, Difei Gu, Gemma E. Moran, Hao Wang, and Dimitris N. Metaxas. Can cross-layer transcoders replace vision transformer activations? an interpretable perspective on vision. *CoRR*, abs/2604.13304, 2026. doi: 10.48550/ARXIV.2604.13304. URL https://doi.org/10.48550/arXiv.2604.13304.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=7uVcpu-gMD.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger B. Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *CoRR*, abs/2209.10652, 2022. doi: 10.48550/ARXIV.2209.10652. URL https://doi.org/10.48550/arXiv.2209.10652.

European Parliament and Council of the European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. Publications Office of the European Union, 2025. doi: doi/10.2804/4225375.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20:177:1–177:81, 2019. URL https://jmlr.org/papers/v20/18-760.html.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 2020. URL http://proceedings.mlr.press/v119/frankle20a.html.

Sven Ove Hansson, Matts-Åke Belin, and Björn Lundgren. Self-driving vehicles—an ethical overview. *Philosophy & Technology*, 34(4):1383–1408, 2021.

Hsiang Hsu and Flávio P. Calmon. Rashomon capacity: A metric for predictive multiplicity in classification. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ba4caa85ecdcafbf9102ab8ec384182d-Abstract-Conference.html.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pp. 2673–2682. PMLR, 2018. URL http://proceedings.mlr.press/v80/kim18d.html.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 10895–10905. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01036.

Dahee Kwon, Sehyun Lee, and Jaesik Choi. Granular concept circuits: Toward a fine-grained circuit discovery for concept representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2356–2365, 2025.

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. Break it down: Evidence for structural compositionality in neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/85069585133c4c168c865e65d72e9775-Abstract-Conference.html.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Maxime Méloux, François Portet, and Maxime Peyrard. Mechanistic interpretability as statistical estimation: A variance analysis of EAP-IG. *CoRR*, abs/2510.00845, 2025. doi: 10.48550/ARXIV.2510.00845. URL https://doi.org/10.48550/arXiv.2510.00845.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *CoRR*, abs/2312.12936, 2023. doi: 10.48550/ARXIV.2312.12936. URL https://doi.org/10.48550/arXiv.2312.12936.

Achyuta Rajaram, Neil Chowdhury, Antonio Torralba, Jacob Andreas, and Sarah Schwettmann. Automatic discovery of visual circuits, 2024. URL https://arxiv.org/abs/2404.14349.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/83004190b1793d7aa15f8d0d49a13eba-Abstract.html.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *IJCNN*, pp. 1453–1460. IEEE, 2011.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders, 2025. URL https://arxiv.org/abs/2502.06755.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2251–2265, 2019. doi: 10.1109/TPAMI.2018.2857768. URL https://doi.org/10.1109/TPAMI.2018.2857768.

Nina Zukowska, Wolfgang Stammer, Bernt Schiele, and Jonas Fischer. Seeing through circuits: Faithful mechanistic interpretability for vision transformers. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision - ECCV 2026 - 19th European Conference, Malmö, Sweden, September 8-12, 2026, Proceedings, Part LXXVIII*, volume 17078 of *Lecture Notes in Computer Science*, pp. 164–181. Springer, 2026. doi: 10.1007/978-3-032-37393-9_11. URL https://doi.org/10.1007/978-3-032-37393-9_11.

A PURITY-BASED RISK SCORE

Motivation. Under $\mathcal{L}_{\text{suff}}$ alone, the mask sees only positive samples. If the positives of a concept are concentrated in a few or a single class, the mask can lower the loss by reproducing the features of the dominant classes regardless of the input. It then fires on any sample, including negatives. This shortcut depends on the data, not on the loss. We therefore measure how concentrated the class labels are among the positives of c, and use the result to scale the specificity weight $\gamma(c)$.

Definition. Let $p_c(k) = |\{(x, c, y) \in \mathcal{D}_c^+ : y = k\}| \,/\, |\mathcal{D}_c^+|$ be the share of class k among the positives of c, computed on the training split. The risk score is the share of the most frequent class:

$$r(c) \;=\; \max_k p_c(k) \;\in\; \Big[\tfrac{1}{|\mathcal{Y}_c|},\, 1\Big], \qquad \gamma(c) \;=\; \gamma_0 \cdot r(c), \tag{5}$$

where $\mathcal{Y}_c$ is the set of classes that contain c-positive samples and γ_0 is a single global hyperparameter. In our experiments, we use $\gamma_0 = 3$ for all datasets and architectures. A concept present in one class only has $r(c) = 1$. A concept spread evenly over many classes has a small $r(c)$ and a small γ. The score uses only the labels and the concept annotations. It needs no trained model and is computed before training.

Per-concept values. Table 2 lists $r(c)$, $|\mathcal{Y}_c|$ and the resulting $\gamma(c)$ for every concept in our roster.

Table 2: Number of positive samples $|\mathcal{D}_c^+|$, number of classes with positives $|\mathcal{Y}_c|$ and the Risk score $r(c)$ for each concept studied in this work.

Dataset	Concept	$\lvert\mathcal{D}_c^+\rvert$	$\lvert\mathcal{Y}_c\rvert$	$r(c)$
AwA2	stripes	2747	4	0.426
	slow	15133	20	0.094
	plains	17688	20	0.093
	vegetation	21823	26	0.075
	big	26535	31	0.062
	newworld	29473	41	0.056
	furry	29666	39	0.055
	chewteeth	31030	39	0.053
	quadrapedal	33616	43	0.049
GTSRB	shape_square	2100	1	1.000
	border_color_white	2880	2	0.729
	symbol_pedestrian	780	2	0.692
	number_1	2850	2	0.505
	main_color_blue	5639	8	0.367
	shape_triangular	11130	16	0.194
	number_0	13200	9	0.170
	shape_round	25199	25	0.089
	border_color_red	28440	28	0.079
	main_color_white	29580	32	0.076

B CONCEPT SELECTION

Statistics. For every concept we compute, directly from the class-level annotations and without training anything, (i) N_{classes}, the number of classes in which the concept is present, (ii) n, the number of concept-positive samples (GTSRB training split, all AWA2 images), and (iii) purity, the fraction of concept-positive samples that belong to the most frequent class among them. Purity matters because a mask trained on positive samples only can lower its loss by ignoring the input and always predicting the dominant class. A concept with purity 1 or few samples is therefore at risk of collapse, and the roster has to include such concepts on purpose so that we can see what happens to them.

Selection. We inventoried all 43 GTSRB concepts and all 85 AWA2 attributes and chose the roster before training, aiming to (a) cover every concept group of GTSRB and as many attribute groups of AWA2 as possible, and (b) span N_{classes}, purity and n as widely as the data allow rather than cluster in the low-purity, many-class corner where most AWA2 attributes fall. We then trained candidate concepts and revised the AWA2 roster where training failed (below).

GTSRB (10 concepts). Table 3 lists them. The three shape concepts, two border colors and two main colors form a controlled ladder from one class to more than 25 classes. Digits and one pictogram add symbolic concepts at low and mid purity.

Table 3: GTSRB roster. N_{classes}: classes containing the concept; purity: share of the most frequent class among positive samples.

Concept	Group	N_{classes}	Purity	n	Role
`shape_square`	shape	1	1.000	2 100	single-class geometric baseline
`shape_triangular`	shape	16	0.194	11 130	mid-N geometric
`shape_round`	shape	25	0.089	25 199	high-N geometric
`border_color_white`	border color	2	0.729	2 880	perceptual, high purity
`border_color_red`	border color	28	0.079	28 440	perceptual, high N, low purity
`main_color_blue`	main color	8	0.367	5 639	perceptual, mid purity
`main_color_white`	main color	32	0.076	29 580	largest N, low purity
`number_1`	number	2	0.505	2 850	symbolic baseline
`number_0`	number	9	0.170	13 200	symbolic, low purity
`symbol_pedestrian`	symbol	2	0.692	780	symbolic, small n

AWA2 (9 concepts). Table 4 lists the final roster. AWA2 attributes are mostly low purity (below 0.15 for most of the 85), so this roster sits in a different regime from GTSRB: purity 0.049–0.426 and N_{classes} between 4 and 43. It includes visual attributes (texture, size), morphological ones (teeth), and non-visual ones (behavior, diet, habitat, taxonomy).

Table 4: AWA2 roster after validation training.

Concept	Group	N_{classes}	Purity	n	Type
`stripes`	texture	4	0.426	2 747	visual, highest purity
`furry`	texture	39	0.055	29 666	visual, many classes
`big`	shape/size	31	0.062	26 535	morphological
`chewteeth`	parts	39	0.053	31 030	morphological
`quadrapedal`	behavior	43	0.049	33 616	non-visual, extreme N and purity
`slow`	behavior	20	0.094	15 133	non-visual, mid N
`vegetation`	diet	26	0.075	21 823	semantic
`plains`	habitat	20	0.093	17 688	semantic, mid N
`newworld`	taxonomy	41	0.056	29 473	semantic, many classes

Concepts dropped after validation training. The first AWA2 roster also contained `flys`, `ground` and `oldworld`. We trained candidates with the validated configuration on ResNet-18 with 10 seeds each, and three of them behaved badly.

- `flys` ($N_{\text{classes}} = 1$, purity 1, 383 samples) collapsed outright at the first seed we tried. The behavior group has no other single-class attribute, so it has no replacement and the AWA2 roster has 9 concepts instead of 10.
- `oldworld` (taxonomy, $N_{\text{classes}} = 44$, purity 0.048) collapsed on 2 of 13 seeds, and `ground` (habitat, $N_{\text{classes}} = 41$, purity 0.051) on 2 of 13 as well. This is not one unlucky draw: we confirmed it with extra seeds.

We replaced `oldworld` with `newworld`, the only other taxonomy attribute, and `ground` with `plains`, a habitat attribute at moderate N_{classes} and low purity. Neither collapsed in 10 seeds. The result for `newworld` is worth noting: it sits almost at the same point as `oldworld` in the (N_{classes},

purity) plane (41 vs. 44 classes, purity 0.056 vs. 0.048), yet it is markedly more stable. The instability is therefore not explained by these two statistics alone, and we suspect it depends on which specific classes a concept picks out. We leave this open.

Coverage and limits of the selection. Across both datasets the roster covers N_{classes} from 1 to 43, purity from 0.049 to 1 and n from 780 to 33 616. It does not cover every attribute group of AWA2, and concept labels are inherited from class-level annotations, so a concept is always a union of classes and cannot be separated from the classes that carry it.

C Implementation of mask training

This section gives the details needed to re-implement our masks in PyTorch. Everything is in plain PyTorch (`torch.fx` for graph rewriting); no other library is needed.

C.1 What is masked

The classifier is frozen. Its weights, biases and BatchNorm statistics are never updated (the network stays in evaluation mode during mask training), so the only trainable parameters are the mask scores. Every weight of a *convolutional* layer gets its own scalar mask score: a layer with weight $W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$ has a score tensor s of the same shape. Masks are therefore per weight, not per channel or per filter.

We mask every `nn.Conv2d` in the network, with two exclusions: the 1×1 shortcut convolutions of ResNet (modules whose name contains `downsample`), and the classification head. We do not mask fully connected layers (CNNs), BatchNorm, biases, or activation functions. Table 5 gives the resulting sizes. For ViT-B/16 (Appendix I) the masked layers are instead the linear layers inside the encoder blocks (Section C.6).

Table 5: Masked layers per architecture, for a 43-class head. The mask entries equal the number of maskable weights; the "fraction" is relative to all parameters of the network.

Model	Conv layers in model	Masked layers	Mask entries	Fraction of parameters
ResNet-18	20	17	10.99 M	98 %
DenseNet-121	120	120	6.87 M	98 %
EfficientNetV2-S	170	170	19.99 M	99 %
ViT-B/16	1	72	84.93 M	99 %

C.2 Mask parametrization

For a masked layer with weight W and score tensor s of the same shape, the forward pass uses

$$\tilde{W} \;=\; W \odot \sigma\big(\beta\, s + \epsilon\big), \tag{6}$$

where σ is the sigmoid, β is a global temperature shared by all layers, and ϵ is a constant offset (Appendix E gives its value). The masked weight $\tilde{W}$ replaces W in the convolution; the bias, stride, padding, dilation and groups of the layer are unchanged. All scores start at $s = 0$. The temperature starts at $\beta = 1$ and is multiplied by a constant factor $\rho = \beta_{\text{end}}^{1/(E-1)}$ at the start of every epoch after the first, so that it reaches β_{end} in the last of E epochs. As β grows, the sigmoid approaches a step function and the soft mask becomes binary. After training, the mask is binarized with $m = \mathbf{1}[s \geq 0]$, and all evaluations use this hard mask, i.e. $\tilde{W} = W \odot m$. Because the training-time mask is $\sigma(\beta s + \epsilon)$ while the final one thresholds at $s = 0$, a positive ϵ biases the soft mask towards keeping weights early in training.

C.3 Graph rewriting with `torch.fx`

We insert the masks without changing the model code. The procedure has three steps.

1. *Registry.* A `MaskRegistry` (an `nn.Module`) holds one `ContinuousMask` submodule per masked layer, keyed by the layer name. Each mask owns the score tensor as an `nn.Parameter`. The registry also holds the temperature as a buffer, so that it is part of the module state and can be updated in place. Registering the masks as submodules of one registry means that the optimizer only has to receive `registry.parameters()`, and that the trained masks can be saved as one `state_dict` (this file is the artifact we keep for every run).
2. *Tracing.* We deep-copy the trained classifier and trace it with `torch.fx.symbolic_trace`. We then build a new graph by copying every node, with one change: each `call_module` node that targets a registered layer is replaced by a `call_function` node that calls a helper `masked_forward(module, x, registry, name)`.
3. *Masked forward.* The helper reads the temperature from the registry buffer, computes the mask $\sigma(\beta s + \epsilon)$ for this layer, multiplies it with `module.weight`, and calls `F.conv2d` (or `F.linear`) with the masked weight and the original layer arguments.

The resulting `GraphModule` is a drop-in replacement of the classifier. We set `requires_grad=False` on all original parameters, so only the registry parameters are trainable. Activation modules of type ReLU, LeakyReLU, GELU, Tanh or Sigmoid are followed by a node that can cache their output; we use it only in analyses after training, and it is inactive during mask training. The hard mask can also be applied without FX by multiplying the stored weights (or with `torch.nn.utils.prune.custom_from_mask`), which we use for analyses that do not need gradients.

Two practical points. (i) The registry is attached to the `GraphModule` only after it has been built, and not to the model before tracing, to avoid reference cycles between the model and the registry during tracing. (ii) The temperature has to be read at run time from the buffer. A Python float captured in the graph would be frozen at trace time and the annealing would have no effect.

C.4 Training objective

A mask is trained for one concept c. Let D^+ be the training images that show the concept and D^- the ones that do not, f_θ the frozen classifier, $\phi(x)$ its *pre-classifier feature vector* (the input of the final linear layer), and P^+ the set of classes in which the concept occurs. The masked network is $f_{\theta \odot m}$ with feature ϕ_m. The loss is

$$\mathcal{L} \;=\; \underbrace{\mathbb{E}_{x \in D^+}\!\left[\tfrac{1}{d}\,\|\phi_m(x) - \phi(x)\|_2^2\right]}_{\text{sufficiency (feature matching)}} \;+\; \gamma\, \underbrace{\mathbb{E}_{x \in D^-}\!\left[\sum_{k \in P^+} \operatorname{softmax}(f_{\theta \odot m}(x))_k\right]}_{\text{specificity}} \;+\; \lambda \sum_{\ell} \left\|\sigma(\beta s_\ell + \epsilon)\right\|_1 \tag{7}$$

with d the feature dimension (the mean squared error, as implemented by `F.mse_loss`). The first term asks the masked network to reproduce the features of the dense network on concept images. The second term is a differentiable relaxation of specificity: on images without the concept, the masked network should put no probability on any class that carries the concept. Without it, a mask on a concept that occurs in only one class can satisfy the first term by always predicting that class. The third term is an ℓ_1 penalty on the soft masks, summed over all layers, and it produces the sparsity. The weight of the specificity term is set per concept as $\gamma = \gamma_0\, r(c)$ with $\gamma_0 = 3$ and $r(c)$ the largest share of one class among the training images that show the concept (the "purity" of the concept, computed from the training split before any mask is trained). Concepts that are nearly identical to a class thus get a strong specificity term, and concepts spread over many classes a weak one. These are the only three terms.

Batches. Because the specificity term needs negatives, training batches are balanced: a sampler alternates positives and negatives, so that a batch of 64 has 32 of each. The number of pairs per epoch is limited by the rarer side, and negatives are re-drawn every epoch. The validation and test splits stay unbalanced. The images of one concept are those of the dataset's own concept annotation (85 binary attributes for AWA2; the class-to-concept table for GTSRB).

C.5 Optimization

Only the scores are optimized, with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay) at a base learning rate η_m (Appendix E). The rate is multiplied by 0.1 at epochs 56 and 71 (84 and 106 for the three-round cells). One epoch is a full pass over the balanced training set. In pseudocode, one round is:

```
s <- 0 for every masked layer; beta <- 1; model.eval()  # frozen backbone
for epoch in 1..E:
    if epoch > 1: beta <- beta * beta_end**(1/(E-1))
    set lr of Adam (schedule above)
    for (x, y) in balanced_train_loader:            # 50% positives
        feat_ref = dense_model.features(x)          # no grad
        logits, feat = masked_model(x)
        loss = mse(feat[pos], feat_ref[pos])
             + gamma * softmax(logits[neg])[:, Y_c].sum(1).mean()
             + lam * sum(sigmoid(beta*s + delta).sum() for s in scores)
        loss.backward(); adam.step()
    validate: feature cosine, accuracy, remaining weights
    early stop on validation feature cosine (see below)
final mask m <- (s >= 0)
```

Early stopping and accelerated finish. We monitor the cosine similarity between the masked and dense pre-classifier features on the validation split. If it has not improved (relative to the mean of the last three epochs) for 20 epochs, we do not stop, but run the remaining epochs with the temperature growing by the per-epoch factor $\rho^{1.2}$ instead of ρ, capped at β_{end}, so that all scores commit to a definite sign; this phase ends early if the cosine falls below 0.8. This reflects that a plateau in fidelity means the soft mask has converged, not that the mask is already binary.

Rounds. Most cells use one round. Three GTSRB DenseNet-121 cells use three rounds; between rounds we reset the temperature to 1 and clamp the scores from above at their initial value, $s \leftarrow \min(s, 0)$ (the "reset" step of continuous sparsification, applied at temperature 1); weights that were kept are thus returned to the decision boundary and the others keep their negative scores. We do not rewind the network weights (they are frozen), so no lottery-ticket rewinding is involved.

Reproducibility. The seed fixes the data split (for AWA2) and the framework RNGs; cuDNN is set to deterministic mode and benchmarking is off. The 30 seeds of a cell differ only in these random choices, not in the frozen classifier.

C.6 Masking the Vision Transformer

`nn.MultiheadAttention` stores the query, key and value projections in one packed `in_proj_weight`, and FX treats the module as a single opaque node. Before tracing, we therefore replace every attention module by an equivalent module with four separate linear layers (`q_proj`, `k_proj`, `v_proj`, `out_proj`), copying the packed weights into them, and we verified numerically that the outputs agree with the original module. We then mask all linear layers of the 12 encoder blocks (the two MLP layers and the four attention projections; 72 layers). The patch-embedding convolution, positional embeddings, LayerNorms and the head are not masked. Everything else is identical to the CNN case.

C.7 Selecting and using the final mask

After the last epoch we save the registry (the scores of all layers) and use its hard binarization. During training, a checkpoint is also kept whenever validation accuracy is within two percentage points of the dense model and the fraction of remaining weights is the lowest so far, but the analyses in the paper use the final registry of each run. A run enters the analysis only if it passes a health check on the accuracy on positives and on the fraction of negatives routed to the target classes, which detects collapsed masks.

Table 6: Hyperparameters of standard training and test accuracy of the classifiers used as the frozen base of all masks. Adam and SGD use the learning rate η and weight decay ω shown; all schedules are cosine annealing to $\eta_{\min} = 10^{-6}$ with period $T_{\max}$. "Backbone" states whether the pretrained layers were updated. Test accuracy is top-1 on the full test split, measured on the checkpoint that the mask runs load. SGD uses momentum 0.9; Epochs: maximum epochs; Pat.: early-stopping patience.

Dataset	Model	Optim.	η	ω	$T_{\max}$	Epochs	Pat.	Backbone	Batch	Acc. (%)
AWA2	ResNet-18	SGD	0.05	5×10^{-4}	100	60	20	trained	64	91.67
AWA2	DenseNet-121	Adam	10^{-4}	10^{-4}	50	50	10	frozen	64	93.77
AWA2	EffNetV2-S	Adam	10^{-4}	10^{-4}	50	50	5	frozen	64	96.25
GTSRB	ResNet-18	SGD	0.05	5×10^{-4}	100	60	20	trained	128	98.19
GTSRB	DenseNet-121	Adam	10^{-3}	10^{-3}	75	20	5	trained	128	98.24
GTSRB	EffNetV2-S	Adam	10^{-3}	10^{-3}	75	20	5	trained	128	98.12

D Architectures and standard training

All masks in the paper are trained on top of a frozen, already trained classifier. This section describes those classifiers and how they were obtained. No concept information is used at this stage: the networks are trained for the plain multi-class task with cross-entropy.

Datasets and splits. *AWA2* has 37,322 images of 50 animal classes. We split it at the image level with a seed-dependent random permutation into 64 % training (23,887 images), 16 % validation (5,971) and 20 % test (7,464); all 50 classes appear in every split. Images are resized to 256 pixels on the short side and cropped to 224×224 (random crop and horizontal flip for training, center crop for evaluation). We use no color jitter, so that color concepts such as *furry* or *stripes* are not altered by augmentation. *GTSRB* has 43 classes. We use the official test set (12,630 images) and split the official training set (39,209 images) into 31,349 training and 7,860 validation images. The split is grouped by physical sign (class and track identifier), because consecutive frames of one track are near-duplicates and a per-image split leaks them across the two sets. Images are resized to 64×64. Training augmentation is a rotation of at most 10°, a translation of at most 5 %, and light color jitter (brightness, contrast, saturation 0.1, hue 0.02). Both datasets are normalized with the ImageNet mean and standard deviation.

Architectures. We use three convolutional backbones from `torchvision`, all initialized with the ImageNet-1k weights (`IMAGENET1K_V1`) and with the final classification layer replaced by a new linear layer with one output per class: ResNet-18 (11.2 M parameters with a 43-way head), DenseNet-121 (7.0 M) and EfficientNetV2-S (20.2 M). No other change is made to the network. In particular, we keep the standard 7×7 stem of ResNet-18 also for the 64×64 GTSRB images. A fourth backbone, ViT-B/16 (85.8 M parameters), is used only in the transformer experiment of Appendix I; it is the `torchvision vit_b_16` with ImageNet-1k weights and a new linear head.

Training procedure. Each classifier is trained once, with seed 42 (the GTSRB EfficientNetV2 checkpoint uses seed 44), and then kept fixed for all concepts and all mask seeds. Training uses the cross-entropy loss and minimizes it with mini-batches of 64 images (AWA2) or 128 images (GTSRB). After every epoch the model is evaluated on the validation split. The checkpoint with the highest validation accuracy is kept, and training stops early if this accuracy has not improved for a number of epochs (the patience). The test split is not used for selection. Table 6 lists the hyperparameters.

On AWA2 the DenseNet-121 and EfficientNetV2-S backbones were kept frozen and only the new linear head was trained; the ResNet-18 and all GTSRB models were fine-tuned end to end.

Performance. All six classifiers reach at least 91.7 % test accuracy (Table 6). The three GTSRB models are within 0.12 percentage points of each other (98.12–98.24 %). On AWA2 the accuracies differ more, from 91.67 % (ResNet-18) to 96.25 % (EfficientNetV2-S), which is expected for a 50-class task with fine-grained, visually overlapping classes. We did not tune the baselines beyond obtaining a strong classifier: the paper studies which parts of a given network carry a concept, not how to improve its accuracy.

ViT-B/16. The ViT-B/16 baselines use the same `vit_b_16` backbone as Appendix I, Adam, weight decay 5×10^{-4}, cosine annealing to $\eta_{\min} = 10^{-6}$, batch size 32, with separate backbone/head learning rates (`src/models/architectures/ViTB16.py`) rather than a single η. On AWA2 the backbone is warm-started from a linear-probe checkpoint (LP-FT) with $\eta_{\text{backbone}} = 10^{-6}$, $\eta_{\text{head}} = 10^{-4}$, $T_{\max} = 6$, 8 epochs, patience 5. On GTSRB the head is randomly initialized and both are fine-tuned from the ImageNet backbone with $\eta_{\text{backbone}} = 10^{-5}$, $\eta_{\text{head}} = 10^{-3}$, $T_{\max} = 30$, 30 epochs, patience 10. Test accuracy: 98.47 % (AWA2), 98.94 % (GTSRB) – in line with the CNN backbones in Table 6.

E MASK TRAINING HYPERPARAMETERS

Table 7 lists the hyperparameters used to train the masks of each of the 57 cells. They were read from the run configuration and log of each of the 30 seeds of a cell; every value is identical across the seeds of a cell, except the risk score r (Appendix A), which is recomputed from each seed's training split and is averaged here. The settings below were shared by all cells. The final threshold is $m_i^{(c)} = \mathbf{1}[\theta_i^{(c)} \geq 0]$, the mask scores are initialised at the offset ϵ, the batch size is 64, and the specificity gain is $\gamma(c) = \gamma_0\, r(c)$ with $\gamma_0 = 3$ in every cell. The temperature starts at $\beta = 1$ and grows geometrically each epoch to β_{end}. Masks were trained in the non-rewinding setting (`ticket=False`), with the mask learning rate dropped by a factor of 10 at the scheduled epochs (56 and 71, or 84 and 106 for the three-round cells).

Hyperparameters are set per architecture, not per concept. Within a dataset–architecture group the settings are the same for every concept, with the exceptions listed below. What varies across groups follows the architecture family: the sparsity weight λ is 3×10^{-7} for ResNet-18 (both datasets), 2×10^{-7} for AWA2 DenseNet-121 and 3×10^{-7} for GTSRB DenseNet-121, and 10^{-9} for GTSRB EfficientNetV2. On AWA2 EfficientNetV2 it is 10^{-7} for five concepts and 5×10^{-8} for four (*big*, *plains*, *slow*, *vegetation*). The mask offset is 2.0 everywhere except GTSRB EfficientNetV2 (0.5) and GTSRB DenseNet-121 *symbol_pedestrian* (1.0). The mask learning rate is 0.01 on AWA2 and 10^{-3} on GTSRB.

Cells trained with a longer schedule. Three GTSRB DenseNet-121 cells (*border_color_white*, *shape_square* and *symbol_pedestrian*) use three pruning rounds of 90 epochs and $\beta_{\text{end}} = 300$; the other 54 cells use one round. Among these, the nine AWA2 EfficientNetV2 cells train for 90 epochs and the rest for 60. All AWA2 cells use $\beta_{\text{end}} = 300$; the remaining GTSRB cells use 200.

Table 7: Mask-training hyperparameters per cell. λ: ℓ_1 weight; ϵ: mask offset; η_m: mask learning rate; epochs are per round; β_{end}: final temperature; r: purity risk score and $\gamma = 3\,r$, averaged over the 30 seeds.

Concept	λ	ϵ	η_m	Epochs	Rounds	β_{end}	r	γ
GTSRB, ResNet-18								
shape_round	3×10^{-7}	2	0.001	60	1	200	0.090	0.271
shape_square	3×10^{-7}	2	0.001	60	1	200	1.000	3.000
border_color_white	3×10^{-7}	2	0.001	60	1	200	0.731	2.192
main_color_blue	3×10^{-7}	2	0.001	60	1	200	0.368	1.104
number_1	3×10^{-7}	2	0.001	60	1	200	0.512	1.536
symbol_pedestrian	3×10^{-7}	2	0.001	60	1	200	0.691	2.072
shape_triangular	3×10^{-7}	2	0.001	60	1	200	0.195	0.584
border_color_red	3×10^{-7}	2	0.001	60	1	200	0.080	0.240
main_color_white	3×10^{-7}	2	0.001	60	1	200	0.077	0.231
number_0	3×10^{-7}	2	0.001	60	1	200	0.173	0.518
GTSRB, EfficientNetV2								
shape_round	1×10^{-9}	0.5	0.001	60	1	200	0.090	0.271
shape_square	1×10^{-9}	0.5	0.001	60	1	200	1.000	3.000
border_color_white	1×10^{-9}	0.5	0.001	60	1	200	0.731	2.192
main_color_blue	1×10^{-9}	0.5	0.001	60	1	200	0.368	1.104
number_1	1×10^{-9}	0.5	0.001	60	1	200	0.512	1.536

Concept	λ	ϵ	η_m	Epochs	Rounds	β_{end}	r	γ
symbol_pedestrian	1×10^{-9}	0.5	0.001	60	1	200	0.691	2.072
shape_triangular	1×10^{-9}	0.5	0.001	60	1	200	0.195	0.584
border_color_red	1×10^{-9}	0.5	0.001	60	1	200	0.080	0.240
main_color_white	1×10^{-9}	0.5	0.001	60	1	200	0.077	0.231
number_0	1×10^{-9}	0.5	0.001	60	1	200	0.173	0.518
GTSRB, DenseNet-121								
shape_round	3×10^{-7}	2	0.001	60	1	200	0.090	0.271
shape_square	3×10^{-7}	2	0.001	90	3	300	1.000	3.000
border_color_white	3×10^{-7}	2	0.001	90	3	300	0.731	2.194
main_color_blue	3×10^{-7}	2	0.001	60	1	200	0.369	1.105
number_1	3×10^{-7}	2	0.001	60	1	200	0.511	1.534
symbol_pedestrian	3×10^{-7}	1	0.001	90	3	300	0.689	2.068
shape_triangular	3×10^{-7}	2	0.001	60	1	200	0.194	0.582
border_color_red	3×10^{-7}	2	0.001	60	1	200	0.080	0.240
main_color_white	3×10^{-7}	2	0.001	60	1	200	0.077	0.231
number_0	3×10^{-7}	2	0.001	60	1	200	0.173	0.518
AWA2, ResNet-18								
quadrapedal	3×10^{-7}	2	0.01	60	1	300	0.049	0.147
slow	3×10^{-7}	2	0.01	60	1	300	0.094	0.281
stripes	3×10^{-7}	2	0.01	60	1	300	0.425	1.274
furry	3×10^{-7}	2	0.01	60	1	300	0.056	0.167
vegetation	3×10^{-7}	2	0.01	60	1	300	0.076	0.227
newworld	3×10^{-7}	2	0.01	60	1	300	0.056	0.168
chewteeth	3×10^{-7}	2	0.01	60	1	300	0.053	0.159
big	3×10^{-7}	2	0.01	60	1	300	0.062	0.187
plains	3×10^{-7}	2	0.01	60	1	300	0.093	0.279
AWA2, EfficientNetV2								
quadrapedal	1×10^{-7}	2	0.01	90	1	300	0.049	0.147
slow	5×10^{-8}	2	0.01	90	1	300	0.094	0.281
stripes	1×10^{-7}	2	0.01	90	1	300	0.425	1.274
furry	1×10^{-7}	2	0.01	90	1	300	0.056	0.167
vegetation	5×10^{-8}	2	0.01	90	1	300	0.076	0.227
newworld	1×10^{-7}	2	0.01	90	1	300	0.056	0.168
chewteeth	1×10^{-7}	2	0.01	90	1	300	0.053	0.159
big	5×10^{-8}	2	0.01	90	1	300	0.062	0.187
plains	5×10^{-8}	2	0.01	90	1	300	0.093	0.279
AWA2, DenseNet-121								
quadrapedal	2×10^{-7}	2	0.01	60	1	300	0.049	0.147
slow	2×10^{-7}	2	0.01	60	1	300	0.094	0.281
stripes	2×10^{-7}	2	0.01	60	1	300	0.425	1.274
furry	2×10^{-7}	2	0.01	60	1	300	0.056	0.167
vegetation	2×10^{-7}	2	0.01	60	1	300	0.076	0.227
newworld	2×10^{-7}	2	0.01	60	1	300	0.056	0.168
chewteeth	2×10^{-7}	2	0.01	60	1	300	0.053	0.159
big	2×10^{-7}	2	0.01	60	1	300	0.062	0.187
plains	2×10^{-7}	2	0.01	60	1	300	0.093	0.279

F PER-CELL VALIDITY RESULTS

This appendix lists the numbers behind the validity figure for all 57 dataset–architecture–concept cells, split by criterion (Tables 8–10); sufficiency and specificity share a table, as do redundancy and sparsity. All values are means over the 30 seeds of a cell.

Table 8: Sufficiency and specificity, per cell (mean ± s.d. over 30 seeds). Sufficiency is the frozen-head AUC of the kept (masked) network against the magnitude-matched null (30 draws; mean and maximum shown); z is the range-based gap of the real mean to the null mean in null standard deviations. Specificity is $1-$ the fraction of concept-negative test inputs predicted as the target. †: real mean does not exceed the null maximum (2/57 cells).

Concept	Kept AUC	Null mean	Null max	z	Specificity
GTSRB, ResNet-18					
shape_round	1.000 ± 0.000	0.517	0.654	7.8	0.997 ± 0.001
shape_square	1.000 ± 0.000	0.964	0.998	1.1	0.999 ± 0.001
border_color_white	1.000 ± 0.000	0.765	0.932	2.2	0.999 ± 0.001
main_color_blue	1.000 ± 0.000	0.644	0.774	3.6	0.999 ± 0.000
number_1	1.000 ± 0.000	0.555	0.761	5.2	0.998 ± 0.003
symbol_pedestrian	0.995 ± 0.009	0.480	0.719	5.3	0.984 ± 0.009
shape_triangular	1.000 ± 0.000	0.453	0.592	8.8	0.999 ± 0.000
border_color_red	0.999 ± 0.000	0.646	0.770	5.2	0.987 ± 0.002
main_color_white	1.000 ± 0.000	0.634	0.772	4.3	0.994 ± 0.002
number_0	1.000 ± 0.000	0.482	0.610	8.6	0.999 ± 0.001
GTSRB, EfficientNetV2					
shape_round	1.000 ± 0.000	0.866	0.971	2.6	0.999 ± 0.001
shape_square	$0.988 \pm 0.006^{\dagger}$	0.979	0.997	0.7	0.999 ± 0.001
border_color_white	$0.981 \pm 0.005^{\dagger}$	0.904	0.989	1.7	0.997 ± 0.003
main_color_blue	1.000 ± 0.000	0.902	0.988	1.3	0.999 ± 0.001
number_1	0.999 ± 0.000	0.937	0.983	1.7	0.997 ± 0.002
symbol_pedestrian	0.999 ± 0.001	0.952	0.989	1.9	0.995 ± 0.006
shape_triangular	1.000 ± 0.000	0.869	0.940	3.2	0.999 ± 0.001
border_color_red	1.000 ± 0.000	0.855	0.944	2.5	0.995 ± 0.003
main_color_white	1.000 ± 0.000	0.824	0.942	2.4	0.996 ± 0.004
number_0	1.000 ± 0.000	0.881	0.967	1.3	1.000 ± 0.000
GTSRB, DenseNet-121					
shape_round	1.000 ± 0.000	0.505	0.679	5.8	0.999 ± 0.001
shape_square	0.996 ± 0.003	0.495	0.634	8.0	1.000 ± 0.000
border_color_white	0.995 ± 0.003	0.522	0.619	7.5	1.000 ± 0.000
main_color_blue	1.000 ± 0.000	0.476	0.923	2.7	1.000 ± 0.000
number_1	1.000 ± 0.000	0.500	0.708	4.8	0.995 ± 0.015
symbol_pedestrian	0.999 ± 0.002	0.494	0.767	4.0	1.000 ± 0.000
shape_triangular	1.000 ± 0.000	0.519	0.657	6.9	1.000 ± 0.000
border_color_red	1.000 ± 0.000	0.488	0.725	4.5	0.995 ± 0.002
main_color_white	1.000 ± 0.000	0.563	0.736	4.0	0.996 ± 0.002
number_0	1.000 ± 0.000	0.491	0.619	8.7	1.000 ± 0.000
AWA2, ResNet-18					
quadrapedal	0.998 ± 0.001	0.901	0.940	2.3	0.984 ± 0.009
slow	0.996 ± 0.001	0.664	0.730	7.3	0.975 ± 0.006
stripes	0.996 ± 0.003	0.812	0.864	4.5	0.995 ± 0.004
furry	0.996 ± 0.006	0.767	0.845	4.0	0.974 ± 0.015
vegetation	0.998 ± 0.001	0.582	0.716	6.8	0.978 ± 0.006
newworld	0.998 ± 0.000	0.625	0.722	7.6	0.989 ± 0.003
chewteeth	0.989 ± 0.006	0.584	0.684	8.0	0.953 ± 0.018
big	0.997 ± 0.001	0.593	0.671	9.5	0.977 ± 0.004
plains	0.996 ± 0.001	0.631	0.721	7.8	0.982 ± 0.003
AWA2, EfficientNetV2					
quadrapedal	1.000 ± 0.000	0.562	0.803	2.9	0.994 ± 0.003
slow	0.999 ± 0.001	0.503	0.589	10.2	0.995 ± 0.001
stripes	0.999 ± 0.001	0.521	0.776	4.3	1.000 ± 0.000
furry	0.999 ± 0.000	0.539	0.679	5.8	0.995 ± 0.002
vegetation	0.999 ± 0.000	0.525	0.619	9.4	0.996 ± 0.001
newworld	0.999 ± 0.001	0.497	0.615	8.9	0.995 ± 0.002
chewteeth	0.998 ± 0.001	0.496	0.595	10.4	0.989 ± 0.006
big	0.999 ± 0.001	0.532	0.601	11.8	0.991 ± 0.003
plains	0.999 ± 0.000	0.502	0.670	6.5	0.997 ± 0.001

Concept	Kept AUC	Null mean	Null max	z	Specificity
AWA2, DenseNet-121					
quadrapedal	1.000 ± 0.000	0.802	0.858	2.7	0.998 ± 0.001
slow	0.999 ± 0.000	0.524	0.584	11.1	0.997 ± 0.001
stripes	1.000 ± 0.000	0.559	0.777	4.6	1.000 ± 0.001
furry	1.000 ± 0.000	0.615	0.733	5.2	0.998 ± 0.001
vegetation	1.000 ± 0.000	0.590	0.682	7.9	0.998 ± 0.001
newworld	1.000 ± 0.000	0.509	0.624	9.8	0.998 ± 0.001
chewteeth	0.999 ± 0.000	0.555	0.651	7.7	0.995 ± 0.003
big	0.999 ± 0.000	0.524	0.592	11.2	0.995 ± 0.002
plains	0.999 ± 0.000	0.567	0.653	5.6	0.998 ± 0.001

Table 9: Necessity, per cell. Frozen-head AUC of the network with the concept mask *inverted* (the kept weights removed), on log-odds scores, mean $\pm$ s.d. over 30 seeds, against the magnitude-matched null (30 draws). Lower is more necessary; 0.5 is chance. $z = (\text{null mean} - \text{real mean})/\hat{\sigma}$, positive when the real inverse AUC is below the null. No pass/fail threshold is applied.

Concept	Inverse AUC	Null mean	Null min	Null max	z
GTSRB, ResNet-18					
shape_round	0.473 ± 0.034	0.514	0.381	0.595	0.8
shape_square	0.525 ± 0.134	0.477	0.305	0.695	-0.5
border_color_white	0.458 ± 0.141	0.474	0.248	0.652	0.2
main_color_blue	0.543 ± 0.191	0.645	0.292	0.868	0.7
number_1	0.522 ± 0.146	0.522	0.271	0.730	-0.0
symbol_pedestrian	0.583 ± 0.104	0.500	0.281	0.718	-0.8
shape_triangular	0.494 ± 0.037	0.482	0.375	0.572	-0.3
border_color_red	0.499 ± 0.080	0.544	0.376	0.696	0.6
main_color_white	0.482 ± 0.073	0.557	0.346	0.735	0.8
number_0	0.438 ± 0.024	0.581	0.432	0.658	2.6
GTSRB, EfficientNetV2					
shape_round	0.498 ± 0.020	0.501	0.470	0.545	0.2
shape_square	0.510 ± 0.088	0.513	0.431	0.647	0.1
border_color_white	0.458 ± 0.077	0.505	0.432	0.632	0.9
main_color_blue	0.465 ± 0.087	0.504	0.392	0.632	0.7
number_1	0.502 ± 0.059	0.491	0.424	0.566	-0.3
symbol_pedestrian	0.491 ± 0.083	0.513	0.403	0.593	0.5
shape_triangular	0.492 ± 0.020	0.497	0.450	0.550	0.2
border_color_red	0.496 ± 0.037	0.500	0.436	0.539	0.1
main_color_white	0.486 ± 0.045	0.500	0.410	0.566	0.4
number_0	0.502 ± 0.030	0.497	0.406	0.544	-0.2
GTSRB, DenseNet-121					
shape_round	0.588 ± 0.073	0.580	0.318	0.745	-0.1
shape_square	0.533 ± 0.074	0.866	0.421	0.997	2.4
border_color_white	0.505 ± 0.103	0.741	0.455	0.973	1.9
main_color_blue	0.370 ± 0.252	0.714	0.315	0.982	2.1
number_1	0.493 ± 0.158	0.528	0.240	0.726	0.3
symbol_pedestrian	0.464 ± 0.138	0.533	0.324	0.851	0.5
shape_triangular	0.527 ± 0.089	0.663	0.317	0.789	1.2
border_color_red	0.555 ± 0.112	0.511	0.212	0.686	-0.4
main_color_white	0.499 ± 0.110	0.552	0.318	0.719	0.5
number_0	0.506 ± 0.070	0.613	0.422	0.812	1.1
AWA2, ResNet-18					
quadrapedal	0.467 ± 0.198	0.475	0.195	0.772	0.1
slow	0.495 ± 0.042	0.511	0.409	0.575	0.4
stripes	0.425 ± 0.120	0.495	0.284	0.744	0.6
furry	0.492 ± 0.095	0.533	0.360	0.678	0.5
vegetation	0.474 ± 0.071	0.494	0.383	0.612	0.3
newworld	0.510 ± 0.030	0.501	0.386	0.606	-0.2

Concept	Inverse AUC	Null mean	Null min	Null max	z
chewteeth	0.494 ± 0.074	0.516	0.407	0.602	0.5
big	0.493 ± 0.029	0.523	0.472	0.584	1.1
plains	0.519 ± 0.076	0.489	0.383	0.656	-0.4
AWA2, EfficientNetV2					
quadrapedal	0.505 ± 0.025	0.499	0.478	0.519	-0.5
slow	0.501 ± 0.006	0.501	0.491	0.509	-0.0
stripes	0.505 ± 0.026	0.498	0.470	0.530	-0.4
furry	0.503 ± 0.009	0.498	0.485	0.511	-0.7
vegetation	0.502 ± 0.008	0.501	0.488	0.513	-0.2
newworld	0.502 ± 0.008	0.502	0.487	0.516	0.0
chewteeth	0.503 ± 0.010	0.503	0.476	0.517	-0.1
big	0.499 ± 0.010	0.501	0.487	0.515	0.2
plains	0.497 ± 0.010	0.498	0.482	0.518	0.1
AWA2, DenseNet-121					
quadrapedal	0.504 ± 0.225	0.562	0.176	0.811	0.4
slow	0.474 ± 0.036	0.498	0.400	0.572	0.6
stripes	0.433 ± 0.086	0.500	0.270	0.754	0.6
furry	0.404 ± 0.104	0.499	0.333	0.658	1.2
vegetation	0.456 ± 0.060	0.492	0.388	0.615	0.7
newworld	0.553 ± 0.046	0.511	0.439	0.624	-0.9
chewteeth	0.472 ± 0.059	0.508	0.398	0.620	0.7
big	0.479 ± 0.028	0.487	0.387	0.560	0.2
plains	0.464 ± 0.070	0.517	0.382	0.652	0.8

Table 10: Redundancy and sparsity, per cell. Redundancy is the feature-projection AUC of the intact, kept network (mean $\pm$ s.d. over 30 seeds) against the shuffled null in the kept condition, which uses only 5 draws (mean and maximum shown). †: real mean does not exceed the null maximum (9/57 cells). Density is the percentage of mask entries kept, mean and min–max over the 30 seeds.

Concept	Kept AUC	Null mean	Null max	Density (%)	Range (%)
GTSRB, ResNet-18					
shape_round	1.000 ± 0.000	0.996	0.997	2.29	2.25–2.34
shape_square	$0.996 \pm 0.007^{\dagger}$	0.999	1.000	5.95	5.70–6.60
border_color_white	$0.998 \pm 0.005^{\dagger}$	0.999	1.000	5.10	4.44–5.79
main_color_blue	1.000 ± 0.000	0.999	1.000	2.22	2.07–2.42
number_1	0.997 ± 0.004	0.981	0.989	1.88	1.72–2.04
symbol_pedestrian	$0.969 \pm 0.042^{\dagger}$	0.977	0.989	4.19	3.73–4.56
shape_triangular	1.000 ± 0.000	0.999	0.999	2.40	2.32–2.57
border_color_red	1.000 ± 0.000	0.996	0.998	2.82	2.73–2.92
main_color_white	1.000 ± 0.000	0.998	0.999	2.97	2.87–3.07
number_0	1.000 ± 0.000	0.991	0.993	1.50	1.48–1.52
GTSRB, EfficientNetV2					
shape_round	1.000 ± 0.000	0.999	1.000	2.03	1.98–2.12
shape_square	$0.981 \pm 0.009^{\dagger}$	0.999	1.000	2.08	1.89–2.31
border_color_white	$0.982 \pm 0.007^{\dagger}$	1.000	1.000	2.40	2.18–2.59
main_color_blue	$1.000 \pm 0.000^{\dagger}$	1.000	1.000	2.42	2.28–2.54
number_1	1.000 ± 0.000	0.995	0.997	2.31	2.13–2.48
symbol_pedestrian	$0.999 \pm 0.001^{\dagger}$	1.000	1.000	2.84	2.63–3.07
shape_triangular	$1.000 \pm 0.000^{\dagger}$	1.000	1.000	2.06	1.96–2.21
border_color_red	1.000 ± 0.000	1.000	1.000	2.21	2.12–2.33
main_color_white	$1.000 \pm 0.001^{\dagger}$	1.000	1.000	2.26	2.20–2.36
number_0	1.000 ± 0.000	0.999	1.000	1.75	1.68–1.89
GTSRB, DenseNet-121					
shape_round	1.000 ± 0.000	0.944	0.964	0.36	0.36–0.36
shape_square	0.984 ± 0.012	0.900	0.940	0.14	0.10–0.35
border_color_white	0.989 ± 0.007	0.909	0.959	0.18	0.13–0.44

Concept	Kept AUC	Null mean	Null max	Density (%)	Range (%)
main_color_blue	1.000 ± 0.000	0.992	0.996	0.26	0.26–0.27
number_1	1.000 ± 0.000	0.840	0.900	0.24	0.23–0.27
symbol_pedestrian	0.979 ± 0.033	0.858	0.907	0.48	0.37–0.63
shape_triangular	1.000 ± 0.000	0.945	0.961	0.31	0.31–0.31
border_color_red	1.000 ± 0.000	0.949	0.981	0.38	0.38–0.38
main_color_white	1.000 ± 0.000	0.954	0.977	0.39	0.38–0.39
number_0	1.000 ± 0.000	0.859	0.892	0.29	0.28–0.29
AWA2, ResNet-18					
quadrapedal	0.998 ± 0.001	0.978	0.982	10.99	9.09–12.54
slow	0.994 ± 0.001	0.881	0.889	8.37	7.68–9.16
stripes	0.996 ± 0.003	0.962	0.967	7.27	6.07–8.51
furry	0.996 ± 0.004	0.951	0.954	9.76	5.11–11.74
vegetation	0.997 ± 0.001	0.874	0.875	8.20	7.72–9.23
newworld	0.998 ± 0.000	0.891	0.895	8.01	7.52–8.66
chewteeth	0.988 ± 0.005	0.864	0.875	11.03	9.20–12.63
big	0.996 ± 0.001	0.880	0.884	7.81	7.23–8.42
plains	0.995 ± 0.001	0.887	0.893	8.59	7.88–9.37
AWA2, EfficientNetV2					
quadrapedal	0.998 ± 0.002	0.964	0.966	3.88	3.76–3.95
slow	0.998 ± 0.001	0.875	0.879	4.64	4.54–4.89
stripes	0.996 ± 0.002	0.948	0.958	3.87	1.47–4.30
furry	0.999 ± 0.001	0.906	0.910	3.03	2.98–3.10
vegetation	0.999 ± 0.001	0.873	0.876	4.32	4.24–4.41
newworld	0.999 ± 0.001	0.867	0.873	3.07	3.03–3.18
chewteeth	0.997 ± 0.003	0.828	0.832	3.43	3.33–3.54
big	0.998 ± 0.001	0.893	0.896	4.98	4.86–5.12
plains	0.998 ± 0.001	0.868	0.870	4.14	4.04–4.28
AWA2, DenseNet-121					
quadrapedal	0.999 ± 0.000	0.960	0.963	10.59	10.30–11.20
slow	0.994 ± 0.000	0.843	0.848	7.98	7.87–8.19
stripes	0.999 ± 0.001	0.938	0.941	8.91	8.58–9.12
furry	0.999 ± 0.000	0.911	0.917	8.74	8.52–9.20
vegetation	0.997 ± 0.001	0.845	0.848	7.79	7.67–8.02
newworld	0.997 ± 0.001	0.864	0.866	9.12	8.91–9.56
chewteeth	0.993 ± 0.001	0.824	0.829	9.75	9.47–10.29
big	0.996 ± 0.001	0.861	0.866	8.29	8.13–8.63
plains	0.995 ± 0.001	0.852	0.856	7.14	7.03–7.32

F.1 CAV Baseline Comparison, Full Results

Following the original CAV protocol of training many probes per concept to account for run-to-run variability, we fit 30 logistic-regression CAVs per cell on the same frozen pre-classifier activations, and separately confirmed a 500-draw random-direction null is at chance for every concept (mean AUC 0.500, all cells within [0.47, 0.53]).

Table 11 reports per-cell detection AUC (mean ± 1 s.d. across seeds) for the linear-probe CAV baseline and the mask-based detector, across all 57 concept-architecture-dataset combinations. The winner column marks a cell *Mask* or *CAV* when the difference in mean AUC exceeds 0.001, and *Tie* otherwise; this margin is a practical threshold on the raw AUC scale, not a significance test.

Table 11: CAV vs. mask detection AUC, all 57 cells.

Concept	Dataset	Backbone	CAV AUC	Mask AUC	Diff	Winner
big	AWA2	DenseNet-121	0.9887 ± 6.9e-4	0.9994 ± 1.9e-4	+0.0107	Mask
big	AWA2	EfficientNetV2	0.9997 ± 1.9e-4	0.9989 ± 5.5e-4	-0.0008	Tie
big	AWA2	ResNet-18	0.9968 ± 2.6e-4	0.9973 ± 6.8e-4	+0.0005	Tie
chewteeth	AWA2	DenseNet-121	0.9602 ± 2.0e-3	0.9988 ± 4.3e-4	+0.0386	Mask

continued on next page

Concept	Dataset	Backbone	CAV AUC	Mask AUC	Diff	Winner
chewteeth	AWA2	EfficientNetV2	0.9990 ± 4.8e-4	0.9982 ± 1.1e-3	-0.0009	Tie
chewteeth	AWA2	ResNet-18	0.9932 ± 6.4e-4	0.9892 ± 6.3e-3	-0.0040	CAV
furry	AWA2	DenseNet-121	0.9965 ± 3.0e-4	0.9998 ± 9.9e-5	+0.0033	Mask
furry	AWA2	EfficientNetV2	0.9999 ± 4.3e-5	0.9993 ± 3.8e-4	-0.0005	Tie
furry	AWA2	ResNet-18	0.9989 ± 1.9e-4	0.9961 ± 5.7e-3	-0.0028	CAV
newworld	AWA2	DenseNet-121	0.9916 ± 7.3e-4	0.9996 ± 1.1e-4	+0.0080	Mask
newworld	AWA2	EfficientNetV2	0.9994 ± 2.9e-4	0.9991 ± 5.2e-4	-0.0003	Tie
newworld	AWA2	ResNet-18	0.9979 ± 2.4e-4	0.9984 ± 2.7e-4	+0.0005	Tie
plains	AWA2	DenseNet-121	0.9854 ± 6.9e-4	0.9991 ± 2.4e-4	+0.0137	Mask
plains	AWA2	EfficientNetV2	0.9996 ± 1.1e-4	0.9989 ± 4.8e-4	-0.0007	Tie
plains	AWA2	ResNet-18	0.9968 ± 1.7e-4	0.9960 ± 8.2e-4	-0.0008	Tie
quadrapedal	AWA2	DenseNet-121	0.9975 ± 2.8e-4	0.9997 ± 1.4e-4	+0.0022	Mask
quadrapedal	AWA2	EfficientNetV2	0.9999 ± 4.8e-5	0.9996 ± 2.9e-4	-0.0002	Tie
quadrapedal	AWA2	ResNet-18	0.9994 ± 8.9e-5	0.9980 ± 1.1e-3	-0.0014	CAV
slow	AWA2	DenseNet-121	0.9809 ± 7.8e-4	0.9989 ± 2.4e-4	+0.0180	Mask
slow	AWA2	EfficientNetV2	0.9985 ± 1.9e-4	0.9988 ± 5.1e-4	+0.0003	Tie
slow	AWA2	ResNet-18	0.9964 ± 2.4e-4	0.9959 ± 1.2e-3	-0.0005	Tie
stripes	AWA2	DenseNet-121	0.9978 ± 5.7e-4	0.9998 ± 1.1e-4	+0.0020	Mask
stripes	AWA2	EfficientNetV2	0.9998 ± 4.7e-4	0.9986 ± 8.5e-4	-0.0011	CAV
stripes	AWA2	ResNet-18	0.9986 ± 5.6e-4	0.9960 ± 2.8e-3	-0.0026	CAV
vegetation	AWA2	DenseNet-121	0.9861 ± 7.2e-4	0.9995 ± 1.4e-4	+0.0134	Mask
vegetation	AWA2	EfficientNetV2	0.9989 ± 1.9e-4	0.9993 ± 3.3e-4	+0.0004	Tie
vegetation	AWA2	ResNet-18	0.9971 ± 1.5e-4	0.9977 ± 8.3e-4	+0.0006	Tie
border_color_red	GTSRB	DenseNet-121	0.9994 ± 2.8e-4	0.9999 ± 7.5e-5	+0.0005	Tie
border_color_red	GTSRB	EfficientNetV2	1.0000 ± 3.2e-6	0.9999 ± 9.2e-5	-0.0001	Tie
border_color_red	GTSRB	ResNet-18	1.0000 ± 7.5e-7	0.9992 ± 4.8e-4	-0.0007	Tie
border_color_white	GTSRB	DenseNet-121	0.9996 ± 1.8e-4	0.9948 ± 2.6e-3	-0.0047	CAV
border_color_white	GTSRB	EfficientNetV2	1.0000 ± 4.2e-6	0.9815 ± 5.1e-3	-0.0185	CAV
border_color_white	GTSRB	ResNet-18	1.0000 ± 5.0e-7	1.0000 ± 3.7e-5	-0.0000	Tie
main_color_blue	GTSRB	DenseNet-121	1.0000 ± 1.2e-5	1.0000 ± 2.4e-5	-0.0000	Tie
main_color_blue	GTSRB	EfficientNetV2	1.0000 ± 1.4e-5	0.9999 ± 1.0e-4	-0.0000	Tie
main_color_blue	GTSRB	ResNet-18	1.0000 ± 1.3e-7	1.0000 ± 2.6e-6	-0.0000	Tie
main_color_white	GTSRB	DenseNet-121	0.9994 ± 2.4e-4	0.9999 ± 4.9e-5	+0.0006	Tie
main_color_white	GTSRB	EfficientNetV2	1.0000 ± 3.2e-6	1.0000 ± 5.2e-5	-0.0000	Tie
main_color_white	GTSRB	ResNet-18	1.0000 ± 4.8e-7	0.9999 ± 2.4e-5	-0.0001	Tie
number_0	GTSRB	DenseNet-121	0.9999 ± 3.7e-5	0.9999 ± 3.0e-5	+0.0000	Tie
number_0	GTSRB	EfficientNetV2	1.0000 ± 5.9e-6	0.9999 ± 6.7e-5	-0.0001	Tie
number_0	GTSRB	ResNet-18	1.0000 ± 1.4e-6	0.9998 ± 5.8e-5	-0.0002	Tie
number_1	GTSRB	DenseNet-121	0.9998 ± 1.1e-4	0.9998 ± 1.5e-4	+0.0001	Tie
number_1	GTSRB	EfficientNetV2	1.0000 ± 7.3e-7	0.9995 ± 4.7e-4	-0.0005	Tie
number_1	GTSRB	ResNet-18	1.0000 ± 9.9e-7	0.9998 ± 1.8e-4	-0.0001	Tie
shape_round	GTSRB	DenseNet-121	0.9998 ± 7.1e-5	1.0000 ± 2.1e-5	+0.0001	Tie
shape_round	GTSRB	EfficientNetV2	1.0000 ± 1.1e-5	1.0000 ± 2.1e-5	-0.0000	Tie
shape_round	GTSRB	ResNet-18	1.0000 ± 4.1e-6	1.0000 ± 8.9e-6	+0.0000	Tie
shape_square	GTSRB	DenseNet-121	0.9995 ± 1.3e-4	0.9957 ± 3.2e-3	-0.0037	CAV
shape_square	GTSRB	EfficientNetV2	1.0000 ± 2.6e-5	0.9882 ± 6.1e-3	-0.0118	CAV
shape_square	GTSRB	ResNet-18	1.0000 ± 0	0.9999 ± 9.8e-5	-0.0001	Tie
shape_triangular	GTSRB	DenseNet-121	0.9997 ± 5.2e-5	0.9999 ± 7.3e-5	+0.0003	Tie
shape_triangular	GTSRB	EfficientNetV2	1.0000 ± 5.2e-6	0.9999 ± 4.3e-5	-0.0001	Tie
shape_triangular	GTSRB	ResNet-18	1.0000 ± 2.0e-6	1.0000 ± 1.2e-5	-0.0000	Tie
symbol_pedestrian	GTSRB	DenseNet-121	1.0000 ± 1.1e-6	0.9990 ± 2.2e-3	-0.0010	Tie
symbol_pedestrian	GTSRB	EfficientNetV2	1.0000 ± 1.3e-6	0.9994 ± 8.7e-4	-0.0006	Tie
symbol_pedestrian	GTSRB	ResNet-18	1.0000 ± 3.7e-7	0.9953 ± 9.5e-3	-0.0047	CAV

G MULTIPLICITY: ADDITIONAL RESULTS

G.1 ENSEMBLE ADMISSION

A seed is admitted to a cell's ensemble if its detection AUC is within ε of the best seed in that cell. At $\varepsilon = 0.01$, 48 of 57 cells admit all 30 seeds; at $\varepsilon = 0.02$, 53 do. The four cells still short at $\varepsilon = 0.02$ are `awa2/resnet18/chewteeth`

and furry (29 seeds each), gtsrb/efficientnetv2/shape_square (28) and gtsrb/resnet18/symbol_pedestrian (29). gtsrb/efficientnetv2/border_color_white is the steepest case, admitting 13 seeds at $\varepsilon = 0.01$ and all 30 at $\varepsilon = 0.02$. The lowest single-seed AUC in the grid is 0.9489 (gtsrb/resnet18/symbol_pedestrian); the reference seed reaches 0.992–1.000 in every cell, and in 48 of 57 cells the worst seed still clears 0.99. Table 12 gives admitted counts and per-cell results for all 57 cells.

Table 12: Per-cell results. Admitted seeds at $\varepsilon \in \{0.01, 0.02, 0.05\}$, structural agreement κ, intersection AUC at k=30 (head detector), class-channel Rashomon capacity, ambiguity on positives and negatives, and number of test positives. Backbones: r18 = ResNet-18, d121 = DenseNet-121, effv2 = EfficientNetV2.

Cell	Admitted			κ	$\text{AUC}_{k=30}$	RC_{class}	Amb.		Test
	0.01	0.02	0.05				pos	neg	pos.
gtsrb/effv2/border_color_white	13	30	30	–	–	1.03	19.4%	1.7%	960
gtsrb/effv2/shape_square	–	28	30	–	–	1.04	20.1%	0.9%	690
gtsrb/r18/symbol_pedestrian	–	29	30	–	–	1.33	29.0%	3.3%	210

G.2 Redundancy of the structural statistics

The determinacy sweep reports five summary statistics per cell: κ, normalized Jaccard, the unanimous-core fraction, union inflation and ρ_{pooled}. They are not five independent pieces of evidence. Across the 57 cells, κ, normalized Jaccard, the core fraction and union inflation form a single cluster: every pairwise Spearman correlation among them is at least 0.944 in absolute value. Reporting all four as separate support would count one underlying quantity several times, so the main text uses κ throughout. ρ_{pooled} is distinct ($|\rho| \leq 0.47$ against the other four) and is treated separately below. The raw, unnormalized intersection and union sizes (D_0, D_{union}) are a different matter: D_0 correlates with κ at only -0.17 but with ρ_{pooled} at $+0.85$, so unnormalized it mostly reflects density and is not used as an agreement measure.

G.3 Weight magnitude

ρ_{pooled} is the pooled correlation between a weight's inclusion in a mask and its magnitude in the backbone. GTSRB masks are weakly magnitude-driven ($\rho \in [0.08, 0.38]$) and AWA2 masks substantially so ($[0.38, 0.63]$). Across cells ρ_{pooled} is negatively related to κ (Spearman -0.47), so agreement is highest where magnitude explains it least; this is reassuring for GTSRB with DenseNet-121, which has the highest agreement and the lowest magnitude dependence. Table 13 gives the group-level values.

Table 13: Structural agreement by group. Median over the cells in each group, with min–max in parentheses. Unanimous core is the all-seed intersection as a fraction of mean single-mask density; union inflation is the all-seed union over the same; ρ is the pooled inclusion–magnitude correlation.

Dataset / backbone	κ	Unanimous core	Union inflation	ρ (magnitude)
GTSRB / DenseNet-121	0.81 (0.65–0.84)	0.65 (0.38–0.70)	1.42× (1.34–1.94)	0.11 (0.08–0.14)
GTSRB / EfficientNetV2	0.71 (0.59–0.80)	0.45 (0.31–0.59)	1.66× (1.42–2.08)	0.32 (0.28–0.38)
GTSRB / ResNet-18	0.58 (0.49–0.63)	0.31 (0.19–0.38)	2.17× (2.03–2.48)	0.32 (0.25–0.34)
AWA2 / DenseNet-121	0.67 (0.61–0.72)	0.45 (0.35–0.52)	1.83× (1.67–1.99)	0.57 (0.50–0.63)
AWA2 / EfficientNetV2	0.52 (0.37–0.56)	0.27 (0.08–0.30)	2.58× (2.40–3.56)	0.44 (0.38–0.48)
AWA2 / ResNet-18	0.56 (0.43–0.62)	0.23 (0.11–0.33)	2.18× (1.97–2.76)	0.53 (0.51–0.59)

G.4 Functional agreement: trajectory details

For each cell we accumulate seeds in five random orderings, at $k \in \{1, 2, 3, 4, 5, 7, 10, 13, 16, 20, 25, 30\}$, and evaluate the running intersection and union with both the head detector and the probe, against null trajectories from uniform and magnitude-matched

masks at matched density. At $k = 1$ the intersection is a single mask and detects at AUC ≈ 1.0 by construction. Table 14 gives group medians for the head detector. The null gap is the intersection AUC at k=30 minus the largest null-intersection AUC at k=30 across both null types and all orderings; positive means outside the null range. Union gain is the union's AUC advantage over the intersection at k=30.

Table 14: Functional agreement by group (medians, head detector).

Dataset / backbone	κ	Intersection AUC$_{k=30}$	Density retained	Null gap	Union gain
GTSRB / EfficientNetV2	0.71	0.954	0.47	$+0.39$	0.03
GTSRB / DenseNet-121	0.81	0.850	0.67	$+0.35$	0.15
GTSRB / ResNet-18	0.58	0.512	0.33	$+0.01$	0.49
AWA2 / DenseNet-121	0.67	0.631	0.46	$+0.13$	0.21
AWA2 / EfficientNetV2	0.52	0.554	0.27	$+0.05$	0.40
AWA2 / ResNet-18	0.56	0.503	0.27	-0.04	0.48

Over the grid, 29 of 57 cells clear the null by more than 0.10 on the head detector, 13 lie between 0 and 0.10 and 15 are at or below zero (12 on the probe). GTSRB with EfficientNetV2 is the cleanest case: all ten cells lie outside the null (gaps 0.32–0.49), and nine of ten hold intersection AUC at or above 0.90 at $k = 30$ while retaining about half the single-seed density. AWA2 with DenseNet-121 separates from the null in five of nine cells, but none reaches 0.90 (best: `stripes` at 0.84).

Agreement against function. Pooled over the 57 cells, κ and the intersection's AUC at k=30 correlate at Spearman $\rho = 0.44$ (Table 15). This association comes from differences between groups. After subtracting each group's median (the six dataset–backbone combinations) from both variables, the pooled correlation is $\rho = -0.05$ (95% bootstrap interval $[-0.32, +0.23]$). Within datasets, the centered correlation is $+0.22$ on GTSRB (interval includes 0) and -0.47 on AWA2 ($[-0.73, -0.09]$), where the raw pooled value is about zero because group offsets cancel a negative within-group trend (Table 15). The AWA2 result matches the negative signs of the per-backbone correlations (Table 18) but is one test among several on cells that share concepts across backbones, so the intervals are optimistic and we treat it as suggestive. Detection AUC is not reliably related to positive-set size (Spearman +0.18, +0.26, 0.59 on GTSRB and +0.07, 0.57, +0.24 on AWA2 for DenseNet-121, EfficientNetV2 and ResNet-18; all intervals include 0), so none of this is affected by the confound in Section G.6.

Table 15: Spearman correlation between κ and AUC$_{k=30}$, raw and after subtracting each group's median (groups are the six dataset–backbone combinations). Intervals are bootstrap 95% over cells; p-values assume independent cells, which the shared concepts across backbones violate.

Set	Version	n	Spearman ρ [95% CI]	p
All	raw	57	$+0.44$ $[+0.21, +0.61]$	0.0006
All	centered	57	-0.05 $[-0.32, +0.23]$	0.69
GTSRB	raw	30	$+0.42$ $[+0.13, +0.63]$	0.020
GTSRB	centered	30	$+0.22$ $[-0.14, +0.55]$	0.25
AWA2	raw	27	-0.01 $[-0.45, +0.38]$	0.95
AWA2	centered	27	-0.47 $[-0.73, -0.09]$	0.014

Most groups fall almost entirely in one quadrant, so quadrant counts describe the groups more than individual cells. For the quadrant split of the main-text figure (at $\kappa = 0.6$ and AUC $= 0.8$), 17 cells have high κ and high AUC (GTSRB EfficientNetV2, 9; GTSRB DenseNet-121, 6; AWA2 DenseNet-121, 2), 21 have low κ and low AUC (AWA2 EfficientNetV2, 9; AWA2 ResNet-18, 6; GTSRB ResNet-18, 6), 15 have high κ and low AUC (7 AWA2 and 4 GTSRB DenseNet-121 cells among them) and 4 have low κ and high AUC. The split is descriptive; several group medians lie close to the $\kappa = 0.6$ line (both ResNet-18 groups), so counts near the boundary are sensitive to it.

Two cautions apply to reading the curves. First, 30 of 57 cells have not stabilized by $k = 30$ on the head detector, so their intersection AUC is still moving at the end of the ensemble. Second, the k at which a curve flattens does not distinguish a cell that has converged on a working core from one that has bottomed out at chance (several ResNet-18 cells flatten early at AUC 0.50), so stabilization

must be read together with the AUC level and null gap. Ordering also matters: 40 of 57 cells show median across-ordering spread above 0.05 at $k \geq 5$, so mean curves describe the ensemble and not any particular accumulation order. As a consistency check across two independent pipelines, the determinacy sweep's unanimous-core fraction and the trajectory's retained density at $k = 30$ measure the same quantity and agree to within 0.02 in five of six groups.

G.5 Behavioral agreement: measures and caveats

The operating threshold for ambiguity and discrepancy is chosen once per cell by maximizing MCC on the validation split, pooled over the real masks, and reused for every seed and null in that cell. Table 16 lists the cells with the highest ambiguity on positives. All are on GTSRB and have small positive sets (210–960 test positives, against a GTSRB median of 2,640).

Table 16: Cells with the highest positive-class ambiguity. Backbones: r18 = ResNet-18, d121 = DenseNet-121, effv2 = EfficientNetV2.

Cell	Amb. pos	Amb. neg	Test pos.	$RC_{concept}$ (all)	RC_{class}
`gtsrb/effv2/symbol_pedestrian`	43.8%	2.9%	210	1.048	1.06
`gtsrb/r18/symbol_pedestrian`	29.0%	3.3%	210	1.086	1.33
`gtsrb/d121/number_1`	21.8%	3.3%	900	1.040	1.22
`gtsrb/effv2/shape_square`	20.1%	0.9%	690	1.020	1.04
`gtsrb/effv2/border_color_white`	19.4%	1.7%	960	1.026	1.03

Class-balance confound. Ambiguity on positives is strongly rank-correlated with the number of test positives: Spearman -0.92, -0.96 and -0.98 on GTSRB and -0.92, -0.65 and -0.35 on AWA2 (DenseNet-121, EfficientNetV2, ResNet-18), and -0.79 pooled over all 57 cells. Cross-cell comparisons of an ambiguity rate therefore confound concept identity with class balance, and we do not read the ambiguity ranking as a ranking of concepts. Class-channel capacity is also related to the number of positives (Spearman -0.21 to -0.78), while concept-channel capacity shows mixed signs. The confound also explains a dataset-level asymmetry: GTSRB ambiguity sits mostly on positives (median 2.6% against 0.9% on negatives) and AWA2 mostly on negatives (3.0% against 5.7%), matching the imbalance (GTSRB: median 2,640 positives against 9,990 negatives; AWA2: 5,341 against 2,123).

Agreement between measures. Pooled over cells, ambiguity tracks concept capacity at Spearman 0.77 and class capacity at 0.50, while discrepancy tracks the others at only 0.16–0.35; these correlations are not adjusted for positive-set size. Thirteen of 57 cells appear in the top fifteen by one family and not the other. Discrepancy is the least stable measure, since normalizing by a reference mask's own error rate inflates the ratio when that rate is near zero, as in most cells, and we therefore treat capacity as primary.

Null ensembles. In 55 of 57 cells the null ensemble is degenerate under our rule (null AUC below 0.7, or predicted-positive rate outside $[0.01, 0.99]$), leaving only `gtsrb/efficientnetv2/number_0` and `shape_triangular` with a usable comparison. In those two the real ensemble is far less diverse than the null (ambiguity 0.5–1.0% against 100%; concept capacity 1.004–1.005 against 1.10–1.24). The degeneracy has two causes. In 40 cells the null masks are near-chance rankers, since at these densities a random mask destroys the backbone. In the remaining 15 the null masks rank well (AUC 0.77–0.98) and are flagged only because the threshold calibrated on the real ensemble puts nearly every null sample on one side of the boundary. A threshold-free capacity comparison on those 15 cells is possible from the saved per-sample scores. The degeneracy rule is a provisional analysis choice, not part of the experiment.

G.6 Positive-set size, structure and behavior

The number of test positives per concept is identical across the three backbones in all 19 (dataset, concept) pairs, so any ranking that depends on it agrees across backbones by construction. It is

strongly related to ambiguity on positives (see above) and, on GTSRB, to κ (Spearman $+0.85$, $+0.93$, $+0.82$ for DenseNet-121, EfficientNetV2 and ResNet-18; AWA2 -0.37, $+0.55$, -0.07, with no consistent sign). Concepts with fewer positives therefore tend to have less agreement between seeds on GTSRB.

Table 17 compares per-concept rankings across backbone pairs (Kendall's τ) before and after removing the effect of positive-set size (both variables rank-transformed and regressed on the rank of log positives; residuals used). The apparent transfer of κ and ambiguity on GTSRB largely disappears. With 9 or 10 concepts per dataset every adjusted interval for these two measures includes 0, so the data are inconclusive rather than showing no transfer; the adjustment can also remove real signal if concept difficulty covaries with prevalence. Because positive counts are shared across backbones, concept identity cannot be separated from evidence size in this data. Intervals are bootstrap 95% over concepts and are given in the accompanying CSV.

Table 17: Kendall's τ between per-concept rankings across backbone pairs (DenseNet–EffNet / DenseNet–ResNet / EffNet–ResNet), raw and after removing positive-set size.

Dataset	Measure	Raw	Adjusted
GTSRB (n=10)	κ	.78 / .82 / .60	.47 / .33 / $-$.02
GTSRB (n=10)	Ambiguity (pos)	.73 / .78 / .78	.16 / .29 / $-$.02
AWA2 (n=9)	κ	.06 / .33 / .17	.17 / .33 / .28
AWA2 (n=9)	Ambiguity (pos)	.72 / .39 / .67	.67 / .39 / .50

Table 18 gives Spearman correlations between κ and function or behavior, per dataset and backbone. We do not report a pooled correlation of κ with behavioral measures across all 57 cells: pooled values are driven by the positive-count confound (on GTSRB the raw association with ambiguity on positives, -0.73 to -0.94, vanishes after adjustment). After adjustment there is no consistent relation between κ and any behavioral measure in either dataset, and κ and AUC are unrelated within GTSRB backbones and negatively related within AWA2 backbones (intervals include 0). Two adjusted values have intervals that exclude 0 (AWA2 ResNet-18, κ against ambiguity on positives and class capacity); they occur on one backbone among about 30 tests and we do not treat them as supported. The tests share concepts and positive counts and are not independent, and with 9 or 10 concepts the power is low, so this is absence of evidence and not evidence of absence.

Table 18: Spearman correlation of κ with function and behavior (DenseNet-121 / EfficientNetV2 / ResNet-18), raw and adjusted for positive-set size. Intervals are in the accompanying CSV.

Dataset	κ against	Raw	Adjusted
GTSRB	$\text{AUC}_{k=30}$	$+.20$ / $+.30$ / $-.40$	$-.08$ / $+.02$ / $+.10$
GTSRB	Ambiguity (pos)	$-.73$ / $-.94$ / $-.81$	$+.16$ / $-.54$ / $-.09$
GTSRB	RC (class)	$-.66$ / $-.26$ / $-.60$	$-.28$ / $-.20$ / $-.49$
AWA2	$\text{AUC}_{k=30}$	$-.67$ / $-.37$ / $-.54$	$-.67$ / $-.17$ / $-.55$
AWA2	Ambiguity (pos)	$+.27$ / $-.40$ / $-.73$	$-.38$ / $-.20$ / $-.75$
AWA2	RC (class)	$-.02$ / $-.18$ / $-.42$	$-.23$ / $+.15$ / $-.68$

G.7 Resolution of structural statistics

Seed-agreement statistics (D_0, D_{union}, κ, normalized Jaccard, the magnitude correlation) are computed exactly for all 57 cells. D_δ is reported at two thresholds that should not be conflated: a proxy threshold calibrated from AUC/AP targets without functional evaluation, available for all 57 cells, and a functional threshold, the point at which held-out concept detection saturates under a forward pass, available for seven spot-checked cells. Unqualified D_δ values refer to the proxy threshold. The full per-weight agreement vector was validated on a two-cell pilot and not extended to the full grid because of storage cost (on the order of 27 GB uncompressed).

H LOCALIZATION

This appendix asks where a valid mask sits inside the frozen network, and whether overlap between two concepts' masks reflects shared circuitry or just co-occurring training images. Section 5 points here for the summary result: seeds agree most in early layers, concept selectivity is highest in late layers, and masks remain concept-selective even in the least reproducible group in the grid.

H.1 SETUP

Per-layer agreement. For two masks restricted to the weights of layer ℓ, with keep densities p_i, p_j and Jaccard index J, we use the chance-corrected agreement

$$\kappa = \frac{J - J_{\text{exp}}}{1 - J_{\text{exp}}}, \qquad J_{\text{exp}} = \frac{p_i p_j}{p_i + p_j - p_i p_j}, \tag{8}$$

where J_{exp} is the Jaccard expected from two independent masks at those densities. A pair where both masks are empty, or both full, is undefined and dropped; a pair where exactly one mask is empty is kept with $\kappa = 0$. All numbers use the uniform30 seed pool: 30 seeds per cell, with identical training configuration within each cell.

Intra, inter and the selectivity gap. *Intra-*κ is the mean over the 435 unordered pairs of a concept's 30 seeds. *Inter-*κ with a partner concept is the mean over the 870 ordered seed pairs with different seed numbers. We exclude same-seed-number pairs because concepts trained from the same seed share data order and initialization; in practice this raises agreement very little (median excess of same-seed over different-seed pairs between 0.00001 and 0.007 across the six dataset/backbone groups, 95th percentile at most 0.016). The *selectivity gap* is intra-κ minus the mean inter-κ over partners. We report it in two variants: against all partners, and against *low-overlap* partners only, i.e. concepts whose label Jaccard with the target concept is below 0.30. The second variant separates agreement that comes from concepts appearing in the same images from agreement the images do not explain.

Inclusion rule. A layer counts only if its gap is defined and at least 23% of both its intra and its inter pairs are valid. We fixed this rule before any cross-cell computation and did not change it afterwards. Layers that fall short are marked undefined rather than assigned a value from a partial sample. The masked layers are all Conv2d layers except downsample and head layers: 17 in ResNet-18, 120 in DenseNet-121 and 170 in EfficientNetV2.

H.2 DEPTH PROFILE

Stages. We group layers by each architecture's stage cuts: 5 stages for ResNet-18 (stem, layer1–4), 5 for DenseNet-121 (conv0, then each dense block with its transition) and 8 for EfficientNetV2 (features.0–7). Stage s of S is plotted at relative depth s/S. A stage's agreement is the mean over its passing layers. A stage is marked mostly undefined when fewer than half of its layers pass; this happens in 2 of 342 cell-stages (DenseNet-121 denseblock3 for *border_color_white* and *symbol_pedestrian* on GTSRB, where 26 of 49 layers are empty in at least half the seeds). Since S and stage width differ between architectures, comparisons below are within-architecture unless stated otherwise. Figure 3 shows intra-κ, inter-κ and both gap variants for every stage of every group.

Masks are selective in every stage. Intra-κ exceeds inter-κ in all 340 defined cell-stages of all 57 cells, under both gap variants (Figure 3, Table 19): the shaded band between the solid and dashed lines never closes. The gap is much larger on GTSRB than on AWA2. Removing high-overlap partners changes the GTSRB gap only slightly but roughly doubles it on AWA2, where 78 of 108 concept pairs have label Jaccard ≥ 0.30 (against 27 of 135 on GTSRB).

Agreement peaks early, selectivity peaks late. Intra-κ is highest at the stem in 44 of 57 cells and generally falls with depth (median within-cell Spearman between stage depth and intra-κ from -0.39 to -1.00 in five of the six groups; GTSRB/DenseNet-121 is the exception at $+0.50$, staying at or above 0.80 throughout). But inter-κ is also highest at the stem, so high early agreement reflects features that all concepts use. Inter-κ falls faster than intra-κ, and the gap grows with depth: in

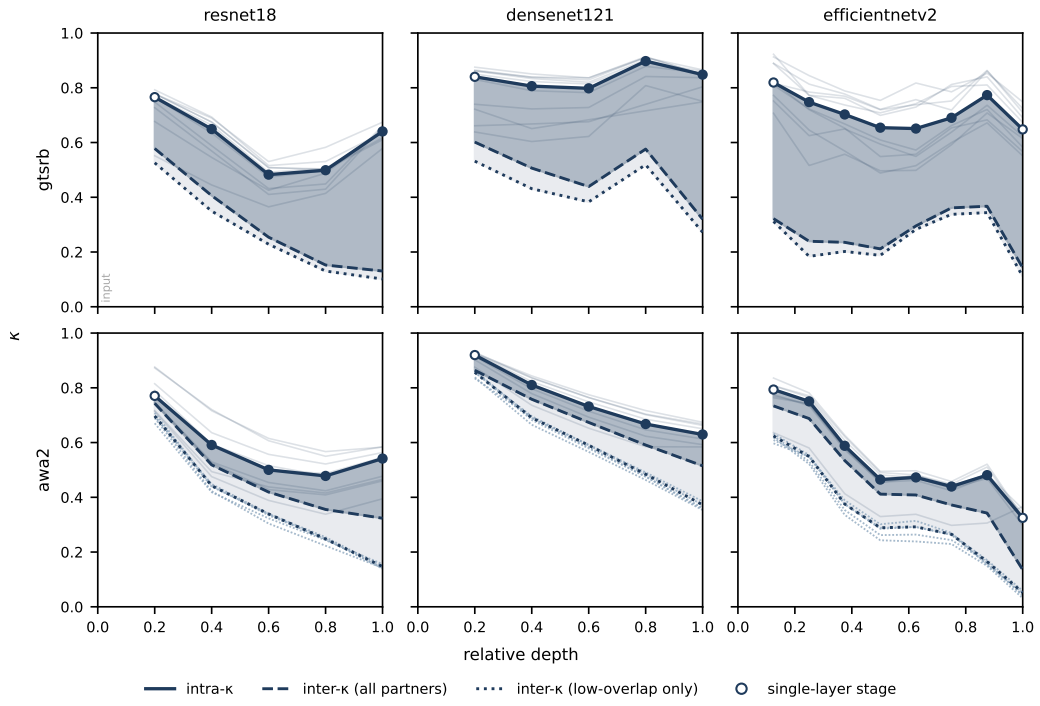


Figure 3: **Concept agreement by relative network depth.** For each (dataset, architecture) pair, the solid line is median intra-concept agreement (κ) across same-concept probes at each depth (thin lines: individual concepts); dashed is median inter-concept agreement against all other probed concepts, dotted against low-overlap partners only. The two shaded bands show the resulting selectivity gap: darker for all partners, lighter for the portion surviving after high-overlap partners are removed. Hollow markers denote single-layer stages (the stem in all architectures; `features.7` in EfficientNetV2 only), which are less robust per our stability check. For five AWA2 concepts with only one usable low-overlap partner at every stage (*big*, *chewteeth*, *newworld*, *quadrapedal*, *vegetation*), the dotted line is drawn per-concept in a desaturated color to flag that it reflects a single comparison rather than a median. The faint line at $x = 0$ ("input") marks the raw input, not a probed layer.

Table 19: Per-cell stage-mean agreement, median across concepts (min–max in parentheses). Gap / intra is the median over cell-stages. The low-overlap gap uses only partners with label Jaccard < 0.30.

Dataset	Backbone	Intra-κ	Inter-κ	Gap	Gap / intra	Gap, low-overlap
GTSRB	EfficientNetV2	0.72 (0.59–0.81)	0.27 (0.20–0.33)	0.44	0.63	0.47
GTSRB	DenseNet-121	0.84 (0.69–0.87)	0.48 (0.28–0.55)	0.33	0.38	0.40
GTSRB	ResNet-18	0.61 (0.47–0.65)	0.30 (0.21–0.37)	0.29	0.50	0.35
AWA2	ResNet-18	0.58 (0.46–0.67)	0.47 (0.37–0.50)	0.11	0.16	0.16
AWA2	EfficientNetV2	0.54 (0.41–0.57)	0.45 (0.32–0.46)	0.09	0.13	0.20
AWA2	DenseNet-121	0.75 (0.69–0.79)	0.68 (0.59–0.69)	0.08	0.09	0.14

Figure 3 the dashed line drops away from the solid one toward the right of each ResNet-18 and DenseNet-121 panel (Table 20).

For DenseNet-121 and ResNet-18 the result is robust: the gap rises with depth in all 38 cells, is largest in the last stage in all 38, and keeps rising after the single-layer stem is dropped, under both gap variants. Masks for different concepts pass through largely the same early weights and separate late. EfficientNetV2 has no consistent ordering. On AWA2 the rise is carried mostly by the final stage, a single layer, and holds in only 5 of 9 cells once that stage and co-occurring partners are removed.

Table 20: Median all-partner gap per stage across concepts, and the median within-cell Spearman correlation between stage depth and gap. The last two columns count cells with a positive correlation, first on all stages, then with single-layer stages (stems, EfficientNetV2 features.7) dropped; the second entry in each is the low-overlap variant.

Dataset	Backbone	Gap by stage, stem → last	Trend	Rising	Rising, dropped
AWA2	DenseNet-121	0.05, 0.06, 0.06, 0.08, 0.13	+1.00	9 / 9	9 / 9, 9 / 9
AWA2	ResNet-18	0.03, 0.07, 0.08, 0.12, 0.24	+1.00	9 / 9	9 / 9, 9 / 9
GTSRB	DenseNet-121	0.23, 0.30, 0.34, 0.31, 0.52	+0.90	10 / 10	10 / 10, 10 / 10
GTSRB	ResNet-18	0.17, 0.23, 0.21, 0.32, 0.51	+0.90	10 / 10	10 / 10, 10 / 10
AWA2	EfficientNetV2	0.06, 0.06, 0.05, 0.05, 0.06, 0.06, 0.15, 0.20	+0.74	9 / 9	8 / 9, 5 / 9
GTSRB	EfficientNetV2	0.52, 0.50, 0.46, 0.43, 0.36, 0.33, 0.42, 0.51	−0.33	0 / 10	0 / 10, 0 / 10

On GTSRB the profile is U-shaped (Figure 3, top right), selective at the stem (0.52), least selective mid-network (0.33) and selective again at the end (0.51). We therefore state the depth result for DenseNet-121 and ResNet-18 only, and describe EfficientNetV2 as selective at every stage without a reliable depth trend.

Concentration. We also computed a concentration statistic C per cell, one minus the normalized entropy of the intra-κ profile across stages. It lies between 0.000 and 0.024 in all 57 cells, so the intra-κ profile is close to flat everywhere, and C closely tracks the squared coefficient of variation of the stage means (Spearman 0.998 across cells). C is strongly anti-correlated with the within-concept κ of Section 5 (Spearman -0.94 overall; -0.91, -0.84 and -0.80 for DenseNet-121, EfficientNetV2 and ResNet-18), but this follows from the definitions: cells with a high mean κ have a smaller relative spread across stages (Spearman -0.95 between mean κ and CV^2). We therefore leave C out of the main claims. The claim that does hold is narrower: selectivity survives in the least reproducible group (AWA2/EfficientNetV2, median κ 0.52, positive gap in every stage), so a mask can be specific to its concept in *where* it sits without being reproducible in *which* weights it keeps.

H.3 Sharing versus co-occurrence

Two concepts' masks can overlap because they share circuitry, or because the images that contain them tend to be the same. Within each dataset and backbone, a pair's mean inter-κ rises with its label Jaccard (slopes $+0.16$ to $+0.48$, R^2 from 0.35 to 0.80). Co-occurrence therefore explains part of the GTSRB/AWA2 difference, but not all of it: among low-overlap pairs alone, AWA2 masks still agree more across concepts (mean inter-κ 0.26–0.52) than GTSRB masks do (0.21–0.38).

We rank concept pairs by the residual of this fit, i.e. by how much more weight they share than their label overlap predicts. Thirteen pairs have a positive residual and low label overlap in all three backbones. The top ones are *shape_round*/*shape_triangular* ($+0.13$), *number_0*/*shape_triangular* ($+0.09$) and *main_color_blue*/*main_color_white* ($+0.08$) on GTSRB, and *plains*/*slow* ($+0.06$) and *furry*/*slow* ($+0.04$) on AWA2. Several top GTSRB pairs have label Jaccard exactly 0: they never occur on the same sign. The two shape pairs are the most suggestive, since both concepts depend on the sign's outline. This ranking is descriptive.

We tried to test it with a pair-specific null: independent uniform random masks with the same per-seed keep counts, drawn 500 times (a self-check gives a 1.28% false-positive rate at $p \leq 0.01$). Every candidate pair beats this null in 98–100% of layers, with observed κ of 0.25–0.62 against a null of 0.000. The eight pairs with the most negative residuals beat it just as clearly (98.3–100% of layers). Any two trained masks overlap far more than random masks of the same density, because all masks inherit common structure such as weight magnitude and layer geometry. The null therefore cannot say whether one pair shares more than another, and we do not cite it as support. A null that preserves this common structure, for example by resampling from per-weight keep rates estimated on other concepts in the same layer, would be needed.

The 0.30 threshold that defines low-overlap partners was fixed before computing any κ and not tuned. On AWA2 it leaves a single low-overlap partner for five of the nine concepts (*big*, *chewteeth*,

newworld, *quadrapedal*, *vegetation*) and eight for *stripes*, so the low-overlap gap for those five concepts is a comparison with one other concept.

H.4 NEGATIVE LAYERS

Of the 5,104 layers that pass the inclusion rule, seven have a negative all-partner gap with at least 100 valid intra pairs, and six remain negative against low-overlap partners only (Table 21). All are small, the largest being -0.028, and every stage that contains one still has a positive gap. Most have a mechanical explanation:

- **Tie from co-occurrence.** In *big*'s features.4.0 block.1, almost the whole layer is kept in every seed, so agreement saturates (intra-κ 0.3987 against inter-κ 0.3989). The tie comes from partners whose labels overlap with *big*'s; without them the gap is positive.
- **Near-empty layers.** The two *shape_round* SE layers in EfficientNetV2 are empty in 77–80% of seeds and have near-zero baseline weights. Only 21 and 15 of 435 intra pairs have two non-empty masks; they pass the inclusion rule only because pairs with one empty mask count as valid.
- **Near-zero-weight layer.** DenseNet-121 denseblock3 layer13 conv2 is negative for both *shape_round* and *symbol_pedestrian*, with identical κ values in both cells. Its mean $|W|$ is 3.4% of the architecture-wide median, and only a handful of its 36,864 weights are ever kept. Every pair of GTSRB concepts has the same inter-κ at this layer (0.9917), except the *shape_round*/*symbol_pedestrian* pair (0.9834), so its masks follow weight magnitude rather than any concept. The same layer is not inert in AWA2-trained backbones (1.19$\times$ the median), so the effect is specific to the GTSRB backbone.
- **Real but small.** The two *stripes* layers are dense layers in the same EfficientNetV2 block. Their negative gap holds against low-overlap partners alone (label Jaccard 0.02–0.11). This is the only case in the grid where a concept's mask agrees with other concepts' masks more than with its own seeds without co-occurrence to explain it, and the effect is at most -0.019.

Table 21: The seven layer-cells with a negative gap. "Seeds empty" is the share of seeds whose mask is empty at that layer; effv2 = EfficientNetV2, d121 = DenseNet-121.

Cell	Layer	Gap, all	Gap, low-overlap	Seeds empty	Keep density
AWA2/effv2/big	features.4.0 block.1 (dw 3×3)	-0.0003	> 0	0%	0.93
AWA2/effv2/stripes	features.4.0 block.1 (dw 3×3)	-0.019	-0.019	0%	0.88
AWA2/effv2/stripes	features.4.0 block.3 (1×1 proj)	-0.014	-0.014	0%	0.65
GTSRB/effv2/shape_round	features.5.5 SE fc1	-0.014	-0.013	77%	$\approx 10^{-5}$
GTSRB/effv2/shape_round	features.5.5 SE fc2	-0.028	-0.033	80%	$\approx 10^{-5}$
GTSRB/d121/shape_round	denseblock3 layer13 conv2	-0.0074	-0.0070	0%	$\approx 8 \times 10^{-5}$
GTSRB/d121/symbol_pedestrian	denseblock3 layer13 conv2	-0.0074	-0.0074	0%	$\approx 8 \times 10^{-5}$

H.5 LIMITATIONS

Seed sets differ between cells (AWA2 42–71; GTSRB ResNet-18 and EfficientNetV2 42–61 and 110–119; GTSRB DenseNet-121 42–51 and 100–119). Training configuration is identical within each cell, but differences between cells cannot be separated from differences between seed sets. Relative depth s/S is comparable within an architecture but not across architectures. The 0.30 overlap threshold was not tuned and leaves some AWA2 concepts with a single partner. Pairs with one empty mask count as valid, which keeps near-empty layers in the analysis; this affects the two undefined DenseNet-121 stages and the two *shape_round* SE layers, and dropping them leaves every stage-level gap positive. We have no test for whether a specific pair shares more weights than others, since the random-mask null is beaten equally by all trained pairs. Finally, this appendix uses only κ from the determinacy sweep; no claim here depends on D_0 or D_{union}.

I ViT-B/16 as a fourth backbone

This appendix describes how concept masks are applied to a Vision Transformer (ViT-B/16) and reports the validity and multiplicity results for it. The ViT study is smaller than the CNN study: 11 concepts (AWA2: big, chewteeth, plains, quadrapedal; GTSRB: border_color_red, border_color_white, main_color_blue, main_color_white, number_1, shape_square, shape_triangular) and 10 seeds per concept, against 19 concepts and 30 seeds for the CNNs. Nulls therefore use 10 draws (sufficiency, necessity) and 5 draws (redundancy).

I.1 Masking the architecture

The backbone is the torchvision `vit_b_16` with ImageNet-1K weights (224×224 input) and a new linear classification head, which is never masked. As for the CNNs, masks are element-wise: one continuous value per weight, with the same parametrisation, initial value and offset. Masks are registered as follows.

1. `nn.MultiheadAttention` packs the Q, K and V projections into a single `in_proj_weight` and is traced as one opaque node, so the individual projections cannot be masked. Before tracing, every attention module is decomposed into explicit `q_proj`, `k_proj`, `v_proj` and `out_proj` linear layers. A parity test checks that the decomposed forward pass equals the original.
2. A mask is attached to each of the following in all 12 encoder blocks: the two MLP linear layers, and the Q, K, V and attention output projections.
3. Not masked: the patch-embedding convolution, class and position tokens, LayerNorms, and the head.

A Q, K or V mask removes input features of a projection, not attention heads or tokens. The structural unit is therefore still the individual weight, but a weight is more entangled than in a CNN because the residual stream carries every layer's output to all later layers. Layer-wise comparisons with the CNNs, where depth is spatially ordered, are therefore not like-for-like.

I.2 Validity and multiplicity per cell

Table 22 lists the per-cell numbers behind Figure 4. Necessity is not in the table: the log-odds rescoring used for the CNNs is not available for ViT, and the raw inverse AUC is shown in the figure only.

I.3 Analysis

Sufficiency, redundancy, specificity. The kept subnetwork recovers the concept in every cell (AUC ≥ 0.996), while size-matched random masks reach 0.41–0.74 on average (at most 0.92). Redundancy behaves as for the CNNs: the intact kept probe and feature projection agree (gap ≈ 0), and shuffling or ablating the mask lowers the feature-projection AUC by 0.2–0.6. One cell, shape_square, is not separated from the 5-draw shuffled null (0.995 against 0.996). Kept-head specificity is 0.993–1.000, and removing the concept collapses the model in all 11 cells.

Sparsity. ViT masks are considerably denser than the CNN masks: density is 0.23–0.34 for AWA2 and 0.07–0.13 for GTSRB, against about 0.001–0.15 for the CNNs.

Necessity. No conclusion is drawn. The raw inverse AUC lies below the null mean in 6 of 11 cells and below the null minimum in 4 (border_color_red, main_color_blue, main_color_white, number_1), and several real values are below 0.5. Raw AUC cannot tell an inverted, still-informative signal from chance, which is why the CNN analysis uses log-odds rescoring; this remains to be run for ViT.

Behavioural multiplicity. Multiplicity is almost absent. The concept-channel capacity is 1.000–1.011 in all cells (bound 2), and the 10 admitted masks per cell are interchangeable at $\epsilon = 0.01$ in

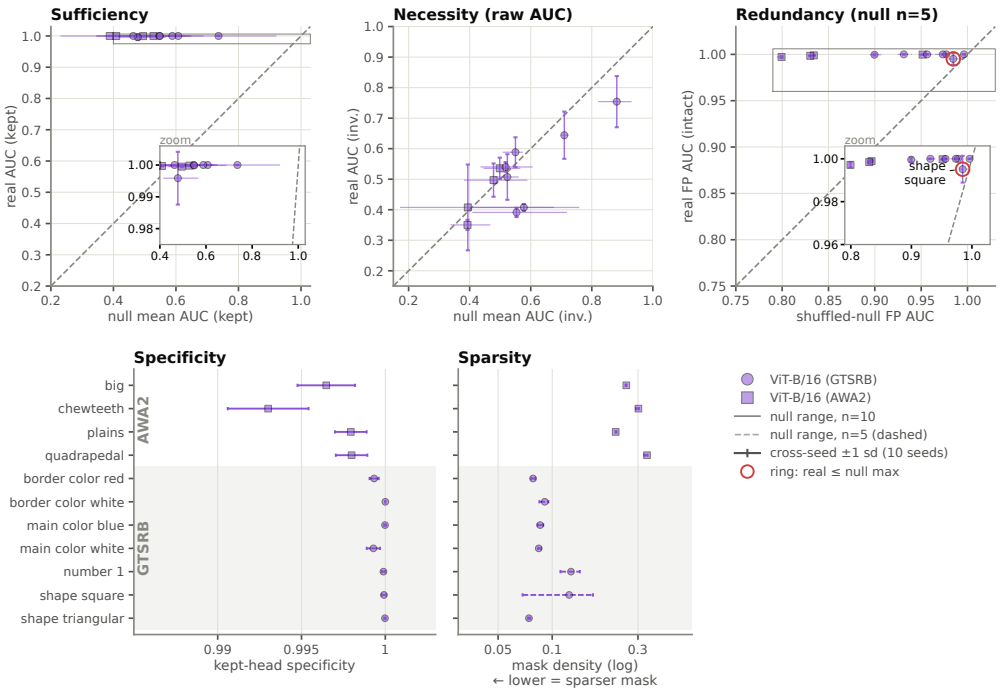


Figure 4: Validity of the concept masks for ViT-B/16 (11 cells, 10 seeds). Sufficiency: real kept AUC against the magnitude null (10 draws); every cell lies above its null range. Necessity uses the raw inverse AUC and is not comparable to the log-odds panel of the CNN figure. Redundancy: real intact feature-projection AUC against the shuffled null (5 draws); the ringed cell, shape_square, is not separated. Specificity and sparsity (mask density, log axis) as in the main figure. Whiskers: ± 1 s.d. over seeds.

10 of 11 cells (shape_square: 9). The class channel is about 1.00 in 8 of 11 cells and larger only for number_1 (3.5), shape_square (3.0) and border_color_white (1.5), all low-prevalence concepts (5–8%). Positive-class ambiguity ranks like the CNNs' (Spearman 0.94 against the CNN median, $n = 11$) but is lower in absolute terms; the class channel does not rank consistently across backbones (Kendall τ 0.0–0.35). The magnitude-null comparison was skipped for these cells, and with 4 AWA2 concepts the per-dataset rank statistics are indicative only.

Structural agreement. The mask κ across seeds is 0.35–0.77 (median 0.60), close to the CNN range (0.37–0.84, median 0.62), and the cell-wise ranking follows the CNN median (Spearman 0.71, $n = 11$); shape_square is lowest (0.35).

Functional agreement. The intersection of all 10 seeds' masks, the core that every seed keeps, still carries the concept in most cells. Its frozen-head AUC is 0.70–0.96 in 7 of 11 cells and 0.57–0.63 in three AWA2 cells (big, chewteeth, plains). It exceeds the maximum of the intersection nulls in 10 of 11 cells, by 0.02 (big) to 0.46 (main_color_blue); the exception is border_color_white (0.48, chance level). For the CNNs the intersection at $k = 30$ spans 0.34–0.99 and exceeds the null in 42 of 57 cells. The relation between κ and intersection AUC is weak for ViT (Spearman 0.50, $n = 11$) and for the CNNs (0.44, $n = 57$). shape_square illustrates the dissociation: it has the lowest κ (0.35) yet an intersection AUC of 0.86. The intersection is taken over 10 seeds for ViT and 30 for the CNNs, and an intersection over more seeds is smaller, so the comparison is indicative only.

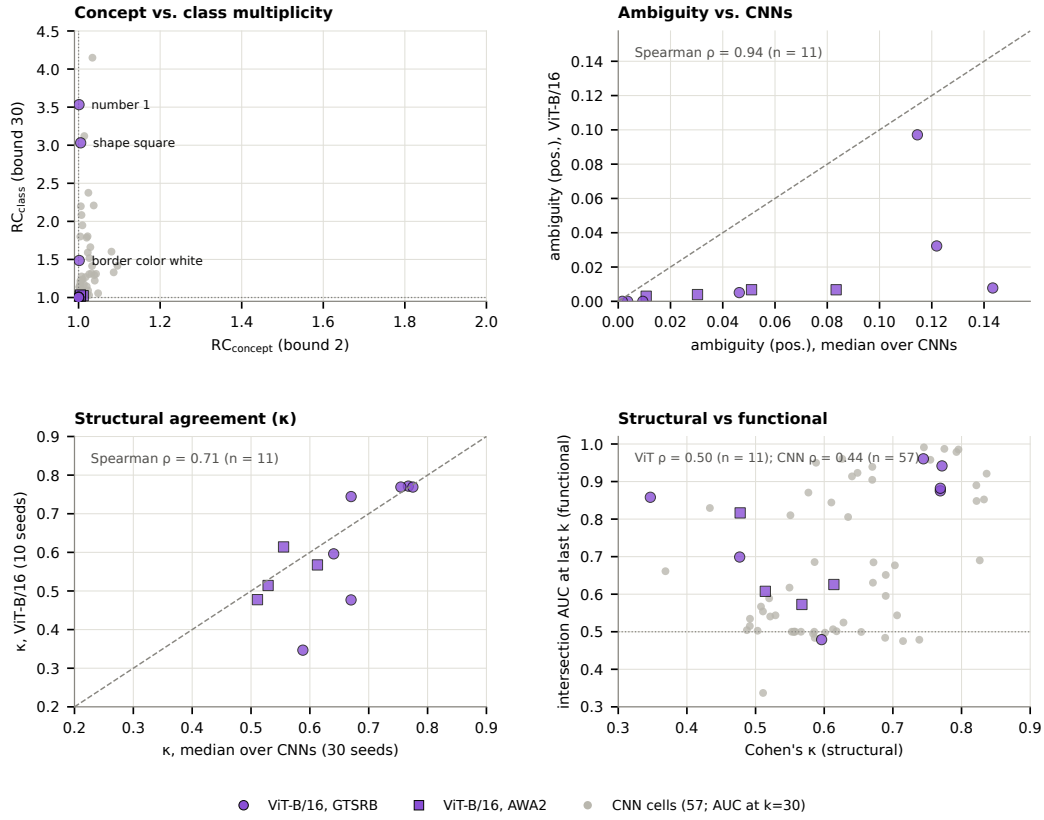


Figure 5: Multiplicity for ViT-B/16. Top left: concept-channel against class-channel Rashomon capacity (grey: the 57 CNN cells). Top right: positive-class ambiguity of ViT against the median over the three CNNs. Bottom left: Cohen's κ of the masks, ViT (10 seeds) against the CNN median (30 seeds). Bottom right: κ against the frozen-head AUC of the intersection of all seeds' masks (ViT: $k = 10$; CNNs, grey: $k = 30$); the dotted line is chance. The seed counts differ, so ViT and CNN values are not exactly comparable.

J SEED-COUNT SENSITIVITY OF THE BEHAVIOURAL RESULTS

This appendix reports how the multiplicity results depend on the number of seeds. It is a descriptive check on a few cells, not a calibration of a stopping rule. All quantities use the 30-seed pool of each cell (57 cells $\times$ 30 seeds, verified clean). Capacity is Rashomon capacity computed with the same routine as in Section 3. Unless stated otherwise, capacity is the mean over all test samples, normalised as $(C-1)/(\min(k,K)-1)$ with $K = 2$ for the concept channel and $K = 30$ for the class channel.

Tier structure at 30 seeds. Both channels are strongly right-skewed. The median normalised capacity is 0.013 (concept) and 0.006 (class). The concept channel has a top group of three cells, `awa2/resnet18/chewteeth` (0.095), `gtsrb/resnet18/symbol_pedestrian` (0.086) and `awa2/resnet18/furry` (0.081), followed by a $1.68\times$ drop to the next cell (0.048). The class channel has a top group of two cells, `awa2/resnet18/stripes` (0.109) and `gtsrb/densenet121/border_color_white` (0.073), followed by a $1.54\times$ drop (0.047). The two groups are disjoint. These groups describe this grid. They are two or three cells each, and we make no claim about a population of cells. We do not set a threshold from them.

Leave-one-out influence (concept channel). For each seed we compute capacity over all 30 seeds minus capacity over the other 29, using the cached per-seed concept scores. Table 23 gives the share of the total contribution carried by the largest one and three seeds, and the within-cell Spearman

Table 22: ViT-B/16, per cell (mean over 10 seeds; ± is the seed s.d.). Suff.: kept-network frozen-head AUC against the magnitude-matched null (10 draws; null mean and max). Red.: intact kept feature-projection AUC against the shuffled null (5 draws; null max). Spec.: kept-head specificity (1− fraction of concept-negative inputs predicted as the target). Dens.: mask density. κ: Cohen's κ of the masks across seeds (10 seeds). †: real mean does not exceed the null maximum (sufficiency); ‡: same for redundancy.

Concept	Suff. AUC	Null mean	Null max	Red. AUC	Null max	Spec.	Dens.	κ
AWA2								
big	1.000 ± 0.000	0.528	0.572	0.999	0.838	0.996 ± 0.002	0.258	0.57
chewteeth	0.999 ± 0.000	0.495	0.550	0.997	0.802	0.993 ± 0.002	0.302	0.51
plains	1.000 ± 0.000	0.409	0.492	0.998	0.832	0.998 ± 0.001	0.226	0.61
quadrapedal	1.000 ± 0.000	0.388	0.631	1.000	0.956	0.998 ± 0.001	0.337	0.48
GTSRB								
border_color_red	1.000 ± 0.000	0.607	0.650	1.000	0.980	0.999 ± 0.000	0.078	0.77
border_color_white	1.000 ± 0.000	0.546	0.638	1.000	0.962	1.000 ± 0.000	0.091	0.60
main_color_blue	1.000 ± 0.000	0.736	0.921	1.000	0.998	1.000 ± 0.000	0.086	0.74
main_color_white	1.000 ± 0.000	0.588	0.649	1.000	0.978	0.999 ± 0.000	0.084	0.77
number_1	1.000 ± 0.000	0.549	0.688	1.000	0.907	1.000 ± 0.000	0.127	0.48
shape_square	0.996 ± 0.008	0.478	0.567	$0.995^{\ddagger}$	0.996	1.000 ± 0.000	0.124	0.35
shape_triangular	1.000 ± 0.000	0.463	0.526	1.000	0.940	1.000 ± 0.000	0.074	0.77

correlation between a seed's contribution and its own test AUC (negative means lower AUC, larger contribution).

Table 23: Leave-one-out influence on concept-channel capacity, five cells. Contribution is capacity(30 seeds) minus capacity(29 seeds without that seed). Backbones: r18 = ResNet-18, d121 = DenseNet-121, effv2 = EfficientNetV2.

Cell	Capacity (30)	Top-1 share	Top-3 share	Spearman (AUC, contrib.)
`awa2/r18/chewteeth`	0.095	41%	53%	−0.76
`gtsrb/r18/symbol_pedestrian`	0.086	18%	43%	0.00
`awa2/r18/furry`	0.081	69%	79%	−0.75
`gtsrb/d121/number_1`	0.040	73%	95%	−0.48
`awa2/r18/slow`	0.037	23%	48%	−0.67

Influence is concentrated in one seed in `furry` and `number_1` and spread over many in `symbol_pedestrian`. Within a cell, lower AUC goes with larger contribution in four of the five cells, but not in `symbol_pedestrian`. The AUCs in `number_1` are all close to 1, so its correlation rests on near-ties. We do not derive a screening rule from this.

Class channel under subsampling. For ten cells we ran the 30 masks on 2000 stratified test points and drew 100 random k-subsets of the seeds for each $k \in \{3, 5, 8, 10, 15, 20, 25\}$, using the same seed subsets in every cell. The cells are the two class-channel top cells, the three concept-channel top cells, `number_1`, `slow`, and three low-capacity cells. The 2000-point subsample keeps at least 200 concept-positive inputs, so the positive share differs by dataset (200 positives on the GTSRB cells, about 1600 on AWA2 `chewteeth` and `furry`). *These values are for calibration only and are not the capacity values reported in Section Sect. 3.4.*

The normalization $(C-1)/(\min(k, K)-1)$ does not remove the dependence on k. Raw capacity grows more slowly than $k-1$, so the normalized value falls as seeds are added in every cell (Table 24). For `stripes` it is 0.453 at $k = 3$, 0.212 at $k = 10$ and 0.106 at $k = 30$. Normalized values at different k are therefore not comparable, and we compare cells only at the same k.

Table 25 counts, out of 100 subsets at each k, how often each class-channel top cell ranks among the top two of these ten cells. These are counts for two named cells, not rates, and two cells cannot support a claim about a population.

Table 24: Mean normalized class-channel capacity over 100 random k-subsets (calibration subsample; the last column is the single 30-seed value on the same subsample). Backbones: r18 = ResNet-18, d121 = DenseNet-121, effv2 = EfficientNetV2.

Cell	k=3	5	8	10	15	20	25	30
awa2/r18/stripes	.453	.329	.236	.219	.162	.132	.121	.106
gtsrb/d121/border_color_white	.075	.086	.069	.076	.060	.060	.060	.062
awa2/r18/slow	.144	.108	.085	.077	.063	.054	.046	.041
awa2/r18/furry	.101	.070	.052	.045	.034	.028	.024	.021
gtsrb/r18/symbol_pedestrian	.041	.033	.025	.022	.017	.014	.012	.011
awa2/r18/chewteeth	.064	.047	.034	.030	.023	.019	.016	.014
gtsrb/d121/number_1	.025	.017	.016	.014	.012	.010	.009	.008
awa2/d121/stripes	.025	.018	.014	.012	.010	.008	.007	.006
gtsrb/r18/main_color_blue	.012	.011	.010	.008	.007	.006	.005	.005
gtsrb/r18/border_color_white	.0016	.0013	.0010	.0008	.0007	.0006	.0005	.0005

Table 25: Subsets (of 100) in which the class-channel top cells rank in the top two of the ten cells at the same k. At $k = 30$ (one subset) both are the top two. Backbones: r18 = ResNet-18, d121 = DenseNet-121, effv2 = EfficientNetV2.

	k=3	5	8	10	15	20	25
awa2/r18/stripes	85	99	100	100	100	100	100
gtsrb/d121/border_color_white	14	33	49	54	47	44	64
both	13	33	49	54	47	44	64

stripes is recovered from five seeds. border_color_white on DenseNet-121 is not: even at 25 seeds it is in the top two in 64 of 100 subsets, and it is usually displaced by awa2/resnet18/slow, whose small-k values are inflated by the normalisation effect above. The estimate for border_color_white is also much noisier (coefficient of variation 2.2 at $k = 3$, 0.34 at $k = 25$, against 0.07 for slow at $k = 25$).

Core verdict under subsampling. We also checked how the verdict on the agreed core depends on the number of seeds, using the intersection trajectories already computed for the multiplicity section (five random seed orders per cell, 57 cells, 285 trajectories). The verdict at k seeds is PASS if the intersection of the first k masks reaches AUC ≥ 0.8, the same descriptive threshold used for the quadrant analysis. At $k = 30$, 21 cells pass (the same for the frozen-head and the frozen-probe detector), so 180 of the 285 trajectories fail. Table 26 reports, for each k, how many trajectories already fail (an early negative), how many of these pass at 30 seeds (false early negatives), and how many of the 180 eventual failures have been caught by k.

Table 26: Early negative core verdicts (frozen-head AUC, 285 trajectories = 57 cells $\times$ 5 orders; 180 fail at 30 seeds). Counts are trajectories, not independent cells.

k	Fail at k	Pass at 30	Caught share of the 180	Pass at k, fail at 30
3	57	4 (7%)	29%	127
5	101	5 (5%)	53%	84
7	134	5 (4%)	72%	51
10	165	7 (4%)	88%	22
13	178	8 (4.5%)	94%	10

An early negative is rarely wrong (4–7% of stops at every k shown), but it is also rare at small k: at three seeds 127 of the 180 eventual failures still pass, and a stopping rule of the form "stop at the first failing core" catches about 29% of them at $k = 3$, 53% at $k = 5$ and 88% at $k = 10$. The frozen-probe detector gives the same picture (36%, 51%, 80%). The core AUC is also not monotone in k: in 194 of the 285 trajectories the AUC rises by more than 0.01 between consecutive k at least once, so the shrinking of the intersection does not by itself guarantee a falling AUC. The verdict at

$k = 3$ also differs between orderings in 13 cells (8 with the probe). A three-seed core is therefore optimistic rather than unreliable in both directions.

What this does and does not show. The number of seeds needed to see a cell's disagreement can be read after the fact from per-seed data already computed. It differs between cells: some are dominated by one seed, some spread over many, and one class-channel top cell is recovered from five seeds while the other is not reliably recovered from 25. Per-seed AUC predicts which seeds contribute in most of the examined cells but not in all, so we do not propose it as a screening rule. The core check uses the existing trajectories and covers all 57 cells. The class-channel and leave-one-out results cover ten and five cells, and we did not derive a threshold or a stopping rule for the class channel. A more principled treatment, for example a species-richness estimator such as Chao1 over clusters of solutions, is future work.