
Beyond Scalars: Concept-Based Alignment Analysis in Vision Transformers

Johanna Vielhaben*
Fraunhofer HHI

Dilyara Bareeva
Fraunhofer HHI

Jim Berend
Fraunhofer HHI

Wojciech Samek†
Fraunhofer HHI &
Technical University of Berlin

Nils Strodthoff‡
Carl von Ossietzky Universität Oldenburg

Abstract

Measuring the alignment between representations lets us understand similarities between the feature spaces of different models, such as Vision Transformers trained under diverse paradigms. However, traditional measures for representational alignment yield only scalar values that obscure *how* these spaces agree in terms of learned features. To address this, we combine alignment analysis with concept discovery, allowing a fine-grained breakdown of alignment into individual concepts. This approach reveals both universal concepts across models and each representation’s internal concept structure. We introduce a new definition of concepts as non-linear manifolds, hypothesizing they better capture the geometry of the feature space. A sanity check demonstrates the advantage of this manifold-based definition over linear baselines for concept-based alignment. Finally, our alignment analysis of four different ViTs shows that increased supervision tends to reduce semantic organization in learned representations.

1 Introduction

Vision Transformers (ViTs) [11] are gaining increased popularity as backbones for various computer vision tasks. There is a large zoo of pre-trained models covering different learning paradigms and supervision levels, with different capabilities [19] and thereby different internal representations. Consequently, understanding and comparing these representations is essential for practitioners when selecting pre-trained models, assessing robustness and generalization properties, and designing fine-tuning protocols. This opens up questions like: Where does the model representation change the most and how? Which concepts (i.e. dominant structures in representation space) are encoded in lower layers vs. upper layers? Does the model encode semantically similar concepts in spatial proximity to each other? How are the representations of model A aligned to that of model B across layers? One way to address these questions is by examining patterns in hidden activations and measuring *representational alignment* [35, 44], yet existing methods often provide only a single scalar to indicate similarity [41], leaving finer details unexamined.

In this paper, we propose a fine-grained alignment analysis based on concepts that structure the latent representation. This gives insights into universal concepts between representations of different layers or models, as well as how a single representation is structured. To achieve concept-based alignment we need solutions for **concept discovery**, and measuring the **alignment** between concept

*Now at Graphcore

†Corresponding author wojciech.samek@hhi.fraunhofer.de

‡Corresponding author nils.strodthoff@uol.de

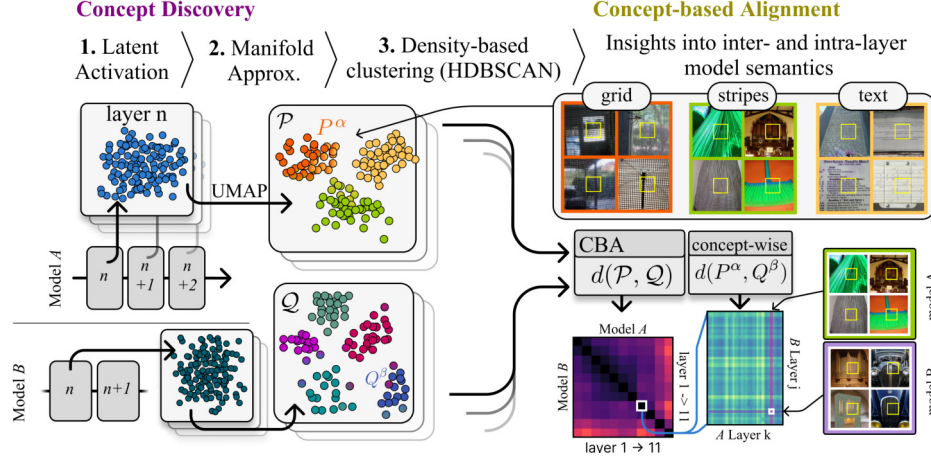


Figure 1: We combine concept discovery with alignment analysis for fine-grained insights into structures within and differences between latent activations. To this end, we investigate latent activations formed by intermediate layers, which according to the manifold hypothesis can be organized in terms of low-dimensional manifolds. We recover these concept manifolds using density-based clustering (HDBSCAN) applied to UMAP embeddings of the latent representations. The discovered structures in latent space allow to characterize a single layer and the formation of structures between layers.

proximity scores. Previous work on **concept discovery** has ranged from identifying individual neurons or other discrete units [1] to defining linear directions in feature space [14], with the most general approach so far viewing concepts as multi-dimensional *linear* subspaces [42]. However, there is growing evidence that linearity may be too strong an assumption [2, 8]. In contrast, we posit that to capture the underlying geometry of a representation more faithfully, concepts should be treated as the most general structure they can form—namely, *nonlinear manifolds*. For **alignment** measurement, previous methods typically characterize the similarity of similarities (e.g., via CKA [28]), collapsing the results into a single scalar score. We instead measure fine-grained distances between concept pairs. Specifically, we represent activations as proximity scores to discovered concepts and employ a generalized Rand index with pseudo-metric properties [24], which we partition into pairwise concept distances.

- We combine **concept discovery** with **alignment** analysis to reveal not only which concepts are universal between two representations but also how a single representation is internally structured (Sec. 2.2).
- We propose a novel definition of concepts as *nonlinear manifolds* and demonstrate—via a sanity check—the superiority of this definition for alignment analysis when compared to simpler and linear concept definitions (Sec. 4.2).
- In our concept-based alignment analysis of different ViTs, we find that their representations exhibit markedly different structures; specifically, increased supervision correlates with reduced semantic structure in the learned representations (Sec. 4.3).

2 Concept Discovery for Representational Alignment

This section consists of two parts. First, we introduce our novel non-linear concept definition – motivated by the manifold hypothesis – and describe our approach to discovering these concepts in latent activations (Fig. 1). Second, we show how this concept-based representation of hidden activations can be used to measure alignment across different models or layers, enabling a fine-grained analysis that goes beyond single-scalar metrics.

2.1 Concept Discovery

Motivation According to the manifold hypothesis, which is widely accepted in machine learning, many datasets, including image data that nominally lie in a high-dimensional space, can be described in terms of a few underlying latent factors and are thus concentrated on a (potentially disconnected) low-dimensional manifold embedded in high-dimensional space [20]. [32] shows how a neural network trained on a toy classification problem solves the task by transforming the topology of the input data, and layerwise reducing the Betti numbers of the class-wise components. We hypothesize that state-of-the-art vision models behave similarly and try to recover the connected components in the hidden representations, which we call *concepts*.

Definition We analyze the hidden representation at an intermediate feature layer of a neural network. To this end, we divide the model f into two parts, $f = g_l \circ h_l$, where h_l is the mapping to a hidden feature layer l . Our definition then relies on hidden representations $h_l(x_i) \in \mathbb{R}^{N' \times F}$ of input samples x_i from a set S . N' is the number of spatially separable elements in the representation, i.e., the number of tokens in a transformer model or the number of superpixels in a convolutional feature map. We spatially decompose the feature maps $h(x_i)$ into a set of $N = N' \cdot |S|$ feature vectors $\phi \in \mathbb{R}^F$. Previously, concepts have been mostly defined as linear structures [14, 46]. The most general linear structure would be affine subspaces, which would already represent an extension compared to the recently considered definition as linear subspaces [42]. In this work, we generalize this idea even one step further and define concepts as manifolds in the F -dimensional feature space.

Definition 1 We define a concept C^α , as a manifold in d -dimensional feature space, represented by a point cloud $\{\phi_j^\alpha\}$ consisting of the feature vectors ϕ_j that lie on the concept manifold with index α .

Benefits of concept manifold definition In the following, we want to compute concept proximity scores by which we measure alignment. Incorrect assumptions about the structure of the concept manifold, e.g., assuming it has no curvature (affine subspaces) or it is spherical and the distance to the manifold can be estimated by the distance to the centroid, directly lead to distorted concept proximity scores and hence to distorted alignment. Later, in a sanity check our definition performs better than linear and non-linear spherical baselines for measuring representational alignment.

Clustering for concept discovery Having established our definition of concepts as manifolds in feature space, we now turn to the challenge of discovering these concepts through clustering. As stated above, we assume that feature vectors $\{\phi_i\}$ from a hidden representation are sampled from a set of low-dimensional concept manifolds $\{C^\alpha\}$. Recovering these concept manifolds in high-dimensional space ($F = 768$ in our experiments) is a challenging clustering problem. Therefore, we revert to density-based clustering on a low-dimensional embedding of the data [18, 22]. For this embedding, we utilize UMAP (Uniform Manifold Approximation and Projection) [29], a dimensionality reduction technique that preserves local and some global structure. Given that we have no a priori knowledge about the number of clusters, we employ HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), which can handle clusters of varying densities [3]. HDBSCAN builds a hierarchy of clusters based on density, represented by a condensed tree, and allows for robust handling of noise, making it suitable for the possibly intricate structure of feature representation spaces. While UMAP does not fully preserve density, its ability to maintain the overall structure of the data makes it a valuable preprocessing step before applying HDBSCAN. We use the HDBSCAN implementation from [30].

Concept proximity scores We leverage soft clustering with HDBSCAN based on the condensed tree which is roughly a density function over the data points to compute fuzzy cluster membership as described in [30], which we formalize in the appendix for the reader’s convenience. It is based on the distance to concept anchor points of each cluster and an outlier score, both derived from the condensed tree. We now have a fuzzy clustering $\mathcal{P}_{\{\phi\}} = \{\mathbf{P}(\phi_0), \dots, \mathbf{P}(\phi_N)\}$ with n clusters, where $\mathbf{P} \in [0, 1]^n$ holds the concept proximity scores of each concept C^α . We interpret the concept proximity scores $P^\alpha(\phi)$ as the probability that a feature vector ϕ belongs to a concept P^α in clustering $\mathcal{P}$. This approach contrasts with previous concept assignment paradigms [14, 42], which often rely on hard clustering, where each feature vector is assigned to a single concept, or linear methods that project onto specific concept directions, limiting the representation to a more rigid framework. In contrast, our soft clustering method allows for nuanced membership scores that reflect the degree

of belonging to multiple concepts. In the following, we refer to our concept discovery method as *NLMCD* (non-linear multi-dimensional concept discovery).

2.2 Concept-based Representational Alignment

We now address the question of measuring representational alignment based on the concept proximity scores derived from fuzzy clustering.

Pseudo-metric between fuzzy clusterings The concepts are at this point characterized by a probabilistic clustering $\mathcal{P}_{\{\phi\}} = \{\mathbf{P}(\phi_0), \dots, \mathbf{P}(\phi_n)\}$, where $\mathbf{P}(\phi_i) = [P^1(\phi_i), \dots, P^n(\phi_i)]$. We want to measure the similarity between two probabilistic clusterings $\mathcal{P}, \mathcal{Q}$ from two different representations to evaluate how aligned their concepts are. For this purpose, we leverage an extension of the pair-based Rand index generalized to fuzzy clusterings proposed in [24]. The original Rand index counts the number of concordant pairs (either two points are paired or not paired both clusterings) and discordant pairs (two points are paired in one clustering but not in the other). The distance between probabilistic clustering $\mathcal{P}, \mathcal{Q}$ is based on a generalized degree of concordance that is based on the *distance between two membership vectors* $d_{ms}(\mathbf{P}(\phi_i), \mathbf{P}(\phi_j))$:

$$d_{cross}(\mathcal{P}, \mathcal{Q}) = \frac{2}{n(n-1)} \sum_{i < j} |d_{ms}(\mathbf{P}(\phi_i), \mathbf{P}(\phi_j)) - d_{ms}(\mathbf{Q}(\phi_i), \mathbf{Q}(\phi_j))| \quad (1)$$

A commonly used choice for the distance d is $d_{ms}(\mathbf{P}(\phi_i), \mathbf{P}(\phi_j)) = 1 - \|\mathbf{P}(\phi_i) - \mathbf{P}(\phi_j)\|_1$ [9]. Finally, we refer to the similarity between two clusterings, derived from the uncovered concepts, as *Concept-Based Alignment* (CBA):

$$\text{CBA} = 1 - d_{cross}(\mathcal{P}, \mathcal{Q}) \quad (2)$$

We choose this measure because $d_{cross}(\mathcal{P}, \mathcal{Q})$ is a pseudo-metric satisfying desirable properties⁴ that ease interpretation. Also, when $\mathcal{P}, \mathcal{Q}$ are crisp partitions, CBA reduces to the original Rand index.

Distance between single clusters In contrast to conventional measures for representational alignment that yield a single scalar value, our approach provides a more nuanced measure of representational alignment by assessing differences between pairs of single clusters. To measure the distance between two clusters P^α, Q^β from two clusterings $\mathcal{P}, \mathcal{Q}$, we decompose the distance in Eq. 1 into the contribution of single concepts P^α, Q^β and measure the *pairwise similarity between the membership scores* of each feature

$$d_{cross}(P^\alpha, Q^\beta) = \frac{2}{n(n-1)} \sum_{i < j} ||P^\alpha(\phi_i) - P^\alpha(\phi_j)| - |Q^\beta(\phi_i) - Q^\beta(\phi_j)|| \quad (3)$$

Due to the absolute value in Eq. 1, summing over all pairs α, β does not yield the total $d_{cross}(\mathcal{P}, \mathcal{Q})$, but by the triangle inequality $\sum_{\alpha, \beta} d_{cross}(P^\alpha, Q^\beta) \geq d_{cross}(\mathcal{P}, \mathcal{Q})$ the sum is an upper bound for the overall distance between two clusterings.

3 Related work

Alignment Representational alignment measures are categorized, with a particular emphasis on Centered Kernel Alignment (CKA) in [28]. CKA evaluates the similarity of similarities, either linearly or under a non-linear kernel. Similarly, [10] measure alignment through the similarities of binary k-nearest neighbor adjacency matrices, which resembles CKA with a narrow Gaussian kernel. Our method relates to CKA in that it condenses these similarities into clusters and subsequently measures the similarity between these clusterings. In parallel to the initial release of this work, [26] introduced a related concept-based alignment measure which relies on class-wise one-dimensional linear concepts. In contrast, we analyze alignment across classes based on non-linear, multidimensional concepts.

Concept discovery Most existing methods model concepts as linear directions [17, 46, 15, 14, 23, 16]. Generalizing this definition, [42] suggest that concepts can be represented more faithfully as multidimensional linear subspaces. While the above approaches operate unsupervised, without access to concept labels, supervised concept discovery methods have explored more flexible geometries:

⁴1) Identity: $d(x, x) = 0$ for all x , 2) Symmetry: $d(x, y) = d(y, x)$ for all x, y , 3) Triangle Inequality: $d(x, z) \leq d(x, y) + d(y, z)$ for all x, y, z .

[39] estimate concept manifolds as multi-dimensional ellipsoids, [7] employ kernel classifiers for nonlinear concept discovery and report improvements over linear methods, and also [8] find evidence for the existence of non-linear features. Unlike these approaches, our main goal in concept discovery is representation summarization for alignment measurement, rather than interpretability or feature enumeration. For this reason, we employ the most general, non-linear concept definition.

Comparison of Vision Models On the one hand, alignment measures such as CKA have been used to compare the representations of various vision models, including ViTs and ResNets trained on different tasks or datasets [6, 5]. This has been combined with the analysis of patterns in attention maps [44, 35, 33], visualization of feature maps [12, 38], and linear probes [38]. On the other hand, downstream performance is analyzed to guide the selection of pre-trained models for transfer learning [27, 19].

4 Results

We evaluate concept discovery in Sec. 4.1, check the superiority of our new concept definition over linear and simplified baselines for concept alignment analysis in Sec. 4.2, and perform a concept-alignment analysis between four ViTs in Sec. 4.3.

4.1 Concept discovery

Experimental setup First, we outline the details of the concept discovery procedure as described in Sec. 2.1 based on UMAP embeddings and HDBSCAN clustering. For concept discovery and later analysis of representational alignment, we use a random subset of 25 % of the ImageNet train set, stratified samples across all 1000 classes. We study four different ViTs [11] with the same architecture (base, patch size 16, input size 224) but different training objectives and training datasets: **FS** [40] trained supervised on ImageNet-1k [37] classification, **CLIP** [34] trained to contrast images and text from WebImageText [34], **DINO** [4] trained on ImageNet-1k to enforce consistency between augmented views of the same image, and **MAE** [21] trained to reconstruct missing pixels of input data from ImageNet-1k. We perform concept discovery separately for the sequence (SEQ) and the CLS token. We extract activations at the last MLP layer of each of the twelve transformer blocks. Due to computational constraints of UMAP and HDBSCAN limiting the number of tokens for clustering, we reduce the 196 SEQ tokens per image to a single representative token, which also facilitates comparison with CLS experiments. We obtain this token by average pooling over a central 4×4 block of tokens, assuming the image center contains more diverse concepts, while peripheral regions may predominantly capture repetitive background elements. Further, for SEQ tokens, we discard the last block as for the considered models only the CLS token in the final layer enters the loss. Hyperparameter tuning for UMAP and HDBSCAN (based on DBCV) and the final settings are detailed in the appendix.

Evaluation We assess concept discovery using four metrics. First, we report the rate of points classified as noise by HDBSCAN. Second, we compute the density-based cluster validity index (DBCV) [31], which compares intra- vs. inter-cluster density and ranges from $[-1, 1]$ (higher is better).⁵ Third, we evaluate how well the UMAP-based clustering preserves distances by calculating the root mean squared error between the original and embedded distance matrices, normalized by the mean pairwise distance (NRMSE). Finally, we test the robustness of our approach by measuring the alignment between two runs – with fixed model weights and input samples, but different initializations for UMAP and HDBSCAN – using CBA from Eq. 2.

Results Turning to the results on embedding and clustering quality in Fig.2, we find that NRMSE remains consistently low across most layers and models. Exceptions include the CLIP CLS representation in layer one and the DINO CLS representation between layers six and eleven, where NRMSE rises. The DBCV scores range between 0.4 and 0.7, indicative of medium clustering quality, yet remain fairly consistent across both models and token types (SEQ vs. CLS). Given the inherent complexity of the clustering task, this level of performance is reasonable, supported by the qualitative

⁵We do not weight the average DBCV by cluster size as proposed in Moulavi et al. [31] so that noise rate and DBCV remain independent.

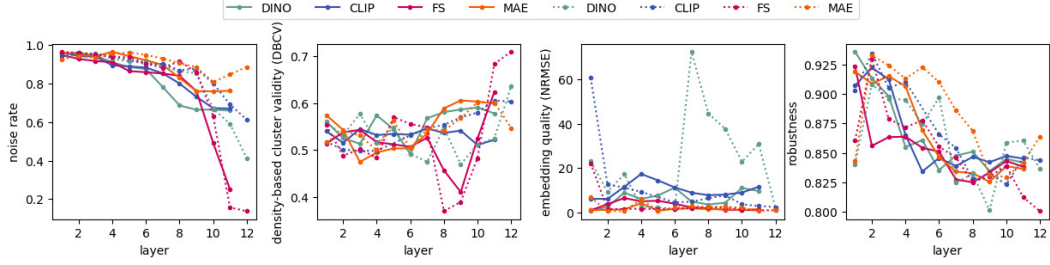


Figure 2: Quality of concept discovery: The noise rate is the ratio of points classified as noise in HDBSCAN. DBCV is a density-based clustering validity index that contrasts intra- vs inter-cluster density with scores in $[-1, 1]$ where higher is better. NRMSE measures the root mean squared error between the distance matrix of the original and embedded activations, normalized by the average distance in the original embedding, and shows how faithfully the UMAP embedding captures the geometry of the representation. Robustness is measured between two runs by concept-alignment from Eq. 2. Results are across layers for CLS (dotted) and SEQ (solid) token representations of the models introduced in Sec. 4.1.

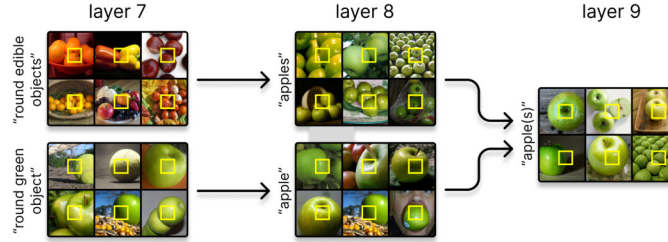


Figure 3: Concept formation graph for the concept “apple(s)” in layer 9 of the FS model. Each concept is represented by six randomly sampled images containing a token assigned to that concept (highlighted in a yellow frame).

results in Fig.3 and Fig. 4. The noise rate is relatively high overall but diminishes in deeper layers. Insufficiently dense sampling, constrained computationally by UMAP and HDBSCAN, may prevent some noisy regions from crystallizing into valid concept clusters. Robustness decreases for all models across layers but stagnates at around 0.84 for most models in the late layers. Finally, two random baselines based on a randomly initialized model and/or randomized input (detailed in the appendix) exhibit weaker clustering validity, confirming that concept discovery relies substantially on both the learned representations and the underlying input distribution.

Concept formation graphs Finally, for a qualitative evaluation of our concept discovery method, we construct concept formation graphs (CFGs). These are unweighted, directed graphs that show how tokens transition from one concept to another across consecutive layers. In Fig. 3, we illustrate how the “apples” concept develops from layer seven to layer nine in the FS model. Additional examples for other models and the detailed algorithm for CFG construction are provided in the appendix.

4.2 Sanity checking concept structure for alignment and CKA comparison

Setup We use a sanity check to demonstrate how concept-based alignment analysis benefits from representing concepts as non-linear manifolds, comparing these results to alternative concept definitions and discovery methods. Additionally, we compare CBA from Eq. 2 against CKA as an established representational alignment measure that offers a single scalar to indicate alignment. This sanity check relies on the assumption by [25] that adjacent layers should align more strongly than distant ones. For each layer, we compute the weighted Kendall’s Tau correlation [43] between alignment scores and layer distances, separately for upstream and downstream layers. We use hyperbolic weights to prioritize alignment between closer layers (whose alignment is more informative) and separate upstream from downstream layers to accommodate different rates of representational change (e.g., layer six may differ more from layer seven than from layer four). Averaging these correlations

Table 1: Sanity check for concept alignment, based on weighted Kendall Tau [43] between alignment and layer distance. We compare the suitability of NLMCD for Concept-Based Alignment (CBA) concepts against other methods: one-dimensional linear subspaces (PCA), multi-dimensional linear subspaces (MCD), and spherical non-linear concepts (KMeans). Additionally, we compare CBA against CKA. Results within the same standard error interval as the top score for each model are **bold** and those CBA results within the same interval as NLMCD-CBA are *italic*. NLMCD consistently outperforms other concept approaches. While NLMCD-CBA and CKA are en-par, CBA offers the advantage of fine-grained concept-based alignment.

	SEQ				CLS			
	FS	CLIP	DINO	MAE	FS	CLIP	DINO	MAE
PCA-CBA	0.91(2)	0.91(3)	0.88(4)	0.84(3)	0.92(2)	0.91(2)	0.78(5)	0.78(4)
MCD-CBA	0.90(4)	0.92(4)	0.85(4)	0.87(4)	0.82(4)	0.73(4)	0.62(5)	0.73(5)
KMeans-CBA	0.94(2)	0.82(4)	0.87(4)	0.98(1)	0.96(3)	0.73(1)	0.89(5)	0.86(4)
NLMCD-CBA	0.97(1)	0.98(1)	0.92(2)	0.98(1)	0.93(2)	0.96(1)	0.91(2)	0.94(2)
CKA	0.98(1)	0.94(2)	0.99(0)	0.99(1)	0.93(2)	0.97(1)	0.89(3)	0.93(3)

across layers verifies whether the alignment measure reflects the expected structural relationships. Using this sanity check, we compare CBA based on NLMCD concepts against one-dimensional linear subspaces discovered by PCA [46, 14], multi-dimensional linear subspaces discovered by MCD [42], and spherical non-linear concepts discovered by KMeans clustering [14]. For the linear subspaces, we project feature vectors onto the concept subspace and clip negative values to zero to obtain soft membership scores (arguing that a vector pointing in the opposite direction of a concept is inactive). For KMeans concepts, we measure concept proximity by the euclidean distance to the cluster centroid. We also normalize concept membership scores $P^{\alpha'} = P^{\alpha} / \sum_{\alpha} P^{\alpha}$ as their sum is required to be less bounded by one $\sum_{\alpha} P^{\alpha} \leq 1$ in Eq. 1. There is no direct way to estimate the number of concepts for PCA, MCD and KMeans, so we use all $F = 768$ components for PCA as a conservative baseline, and the number of concepts discovered by NLMCD for MCD and KMeans discovery.

Results We present the sanity check in Tab. 1 for SEQ and CLS token alignment. We find that for SEQ tokens, NLMCD shows higher scores than linear concepts except for DINO, where PCA can match its performance. Further, only for DINO and MAE simple nonlinear KMeans is en par with NLMCD. Similarly, for CLS tokens, NLMCD mostly outperforms other concept methods, while only for FS can PCA, and for CLIP and MAE can KMeans match its performance. To complement this quantitative evaluation of how well different concept definitions and discovery methods suit alignment measurement, we visualize PCA, MCD, and KMeans concepts and their geometric structure (based on our concept distance measure) in the appendix. By visual inspection, we find that the non-linear NLMCD and KMeans concepts are most coherent within each concept cluster and also show the highest coherence across groups of closely aligned concepts. When comparing to CKA, NLMCD-CBA achieves similar performance for both SEQ and CLS tokens. However, it offers the additional advantage of providing a fine-grained, concept-level view of alignment, unlike the single-scalar nature of CKA.

4.3 Concept Alignment Analysis

We now investigate concept-based alignment described in Sec. 2.2 between representations across layers and models. We structure the analysis into *intra-model* and *inter-model* to answer questions 1) - 4) openend in the introduction. Due to limited space, we focus on SEQ representation and defer the CLS representation analysis to the appendix. First, we analyze how representations are transformed within one model and how they are structured across layers.

Intra-model: Where does the model representation change the most and how? First, we focus on the intra-model alignment heatmaps between SEQ representations across layers measured by CBA from Eq. 2 in the upper row of Fig. 4. In Fig. 5, we complement these findings with additional concept clustering characteristics: alignment of concepts with ImageNet-1k labels, the intrinsic dimensionality of each concept, and the number of concepts per layer. In CLIP, DINO, and MAE, we observe a prominent break between layers one to six and layers six to eleven. The number of

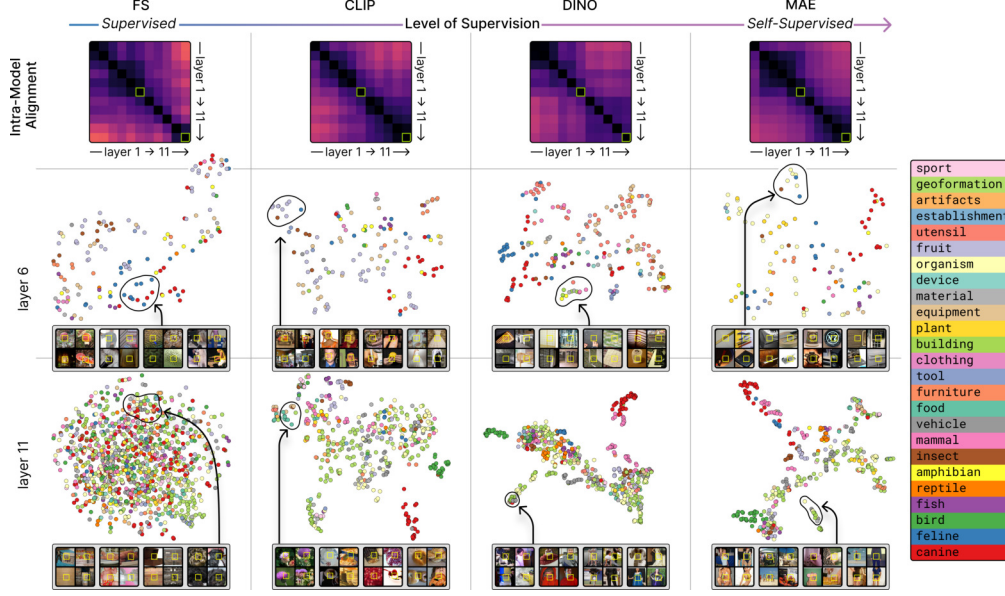


Figure 4: Intra-model relationships of SEQ representations. **Upper row:** We show CBA from Eq. 2 to visualize how representations are transformed across layers of the models introduced in Sec. 4.1 (darker pixels correspond to higher alignment). We observe a nucleation process between layer 9 and 10 in FS and smoother processing split into two major blocks between layer 1-6 and 6-11 in CLIP, DINO and FS. **Center and bottom row:** we zoom into the representations at layer 6 and 11 and partition the scalar CBA alignment into single concepts. We show a UMAP embedding constructed from the pairwise distance of concept measured by $d_{cross}(P^\alpha, P^\beta)$ from Eq. 3. Each point in this *concept atlas* corresponds to a distinct concept P^α . To convey their meaning, for some concepts, we show four random input tokens from the members of the concept cluster P^α marked by a yellow box in the entire image. The stronger the supervision during ViT training ranging from FS, over CLIP to DINO and MAE, the less semantically organized are the representations at layer 11.

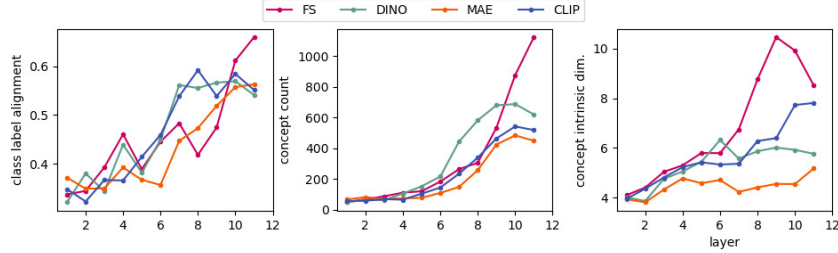


Figure 5: To supplement the intra-model alignment analysis, we evaluate alignment between concepts and ImageNet-1k class labels, (based on CBA from Eq. 2), concept count, and the average intrinsic dimensionality (based on [13]) across concepts.

concepts increases smoothly in the first part but rises sharply from layer seven onward. At that same point, class label alignment shows a marked increase, whereas the average concept dimensionality slightly decreases for DINO and MAE but increases for CLIP. In contrast, FS shows a distinct transformation between layers nine and ten, leading to a sudden jump in both class alignment and the number of concepts, accompanied by higher intrinsic dimensionality. This shift results in a lower alignment between representations in the last two layers and earlier layers and suggests the formation of distinct class-specific concepts, mirroring the “nucleation process” previously reported for ResNets by Doimo et al. [10].

Intra-model: Which concepts are encoded in lower layers vs. upper layers? How structured are the representations? We now zoom in on the representational alignment scores by examining pairwise distances among individual concepts. Specifically, we use a *concept atlas* – a two-dim.

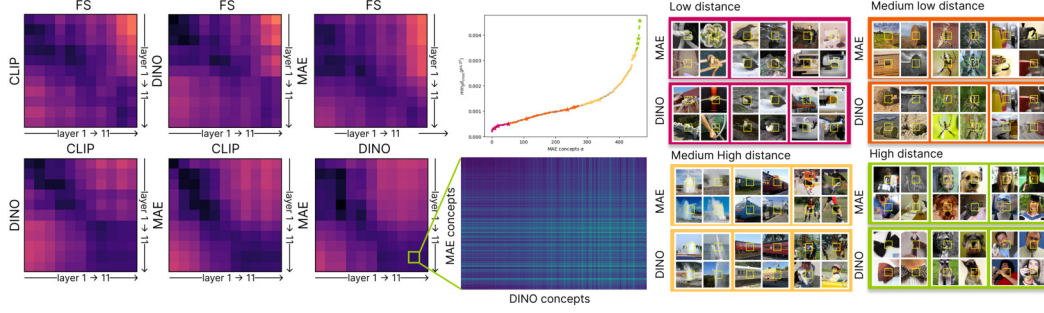


Figure 6: Inter-model relationships of SEQ representations. **Left:** We show CBA from Eq. 2 to visualize how representations differ between models (darker pixels correspond to higher alignment). **Right:** We zoom into the concept-wise distances $d_{cross}(P^\alpha, P^\beta)$ from Eq. 3 between the representation of layer 10 in MAE and DINO. Alongside the pairwise distance matrix $d_{cross}(P^\alpha, Q^\beta)$, we present randomly chosen examples of concepts from MAE and their nearest matches in DINO, and the distribution of the distances between nearest matches. We illustrate some random example pairs with low to high distance (marked by a star in the distance distribution plot). Overall, concepts that show a higher pairwise distance also appear slightly more distinct visually.

UMAP embedding of the pairwise concept distance matrix – to visualize the structures concepts form at layers six and eleven across of all models (Fig. 6). We color-code each concept cluster according to WordNet categories derived from ImageNet-1k labels⁶ to guide the eye. At layer six, all ViT models display a structured set of concepts which mostly focus on lower-level structures, shapes, and object parts, rather than semantic WordNet-based groupings. In comparison, at layer eleven, CLIP, DINO, and MAE concepts show strong semantic organization, illustrated by clearly separated canine concepts and coherent groupings of human body parts such as neck, shoulder, and legs. In contrast, FS at layer eleven appears less semantically organized – fully supervised training may push similar concepts apart (for instance, different dog breeds) to reduce confusion and optimize for task-specific accuracy. This, however, may have negative implications for generalization to other tasks.

Inter-model relations Second, we analyze how the representations between two different models differ and present CBA from Eq. 2 between all layers of the models in the upper part of Fig. 6. We observe higher alignment between the self-supervised models DINO and MAE than with CLIP and the FS model in the alignment heatmaps. Further, layers of the first are more aligned than those of the second half across all models pairs. We conclude that basic foundational features are learned similarly across models, while later layers diverge as the models specialize to concepts serving their pre-training task.

Inter-model: How is the representation of model A similar to that of model B? We now select a pair of layers from two different models (layer ten in MAE and DINO) for a closer look at concept-resolved differences in their representations. Alongside the pairwise distance matrix $d_{cross}(P^\alpha, Q^\beta)$, we present randomly chosen examples of concepts from MAE and their nearest matches in DINO, as well as the distribution of the distances between nearest matches. Overall, concepts that show a higher pairwise distance also appear slightly more distinct visually. Interestingly, around 62% of concepts from MAE all match to only three *singular* concepts in DINO, which are marked in gray in the distance distribution plot in Fig. 6 and excluded from example sampling. Upon inspection, those singularly matched concepts have no visual resemblance. Instead, the singular concepts in DINO exhibit low total activation across samples. Consequently, as our distance measures not only how concepts activate similarly, but also how they similarly do not activate, they are closest match to a variety of MAE concepts that have no close match otherwise – by virtue of their shared lack of activation. Distances to these singular concepts span a range, depending on the activation strength of the concept from the first model. We observe similar behavior for other pairs of layers.

⁶These are derived from the ImageNet-1k labels of the images from which the patches were extracted. We first map these labels to more abstract categories (provided in the appendix). Second, we perform a majority vote among all patches in a cluster to assign the category. This labeling is only a proxy and may not accurately reflect the actual content of the patches—e.g. a patch might show grass on which an animal stands.

5 Conclusion

We propose a novel approach that combines concept discovery with representational alignment analysis in ViTs. With concept-based alignment analysis, we answer the questions raised in the introduction and examine the structure of feature spaces of different ViTs, as well as fine details between the concepts of two different models. These insights are not available through traditional scalar alignment measures. Understanding the structured nature of latent spaces can guide practitioners in choosing models that not only perform well on benchmark datasets but also exhibit robust feature representations for downstream tasks. For instance, the nucleation process in FS emphasizes the importance of model structure over mere classification accuracy when selecting a pre-trained model.

Limitations The computational scalability of HDBSCAN limits the sampling of feature vectors which makes undersampled concept regions appear as noise. The limited variability of ImageNet-1k might obfuscate the meaning of a concept, e.g. when a concept represents a color but there are only dog patches of that color. Further, the computational complexity of CBA requires subsampling of feature vectors.

Acknowledgments This work was supported by the European Union’s Horizon Europe research and innovation programme (EU Horizon Europe) as grants [ACHILLES (101189689), TEMA (101093003)]; and the German Research Foundation (DFG) through research unit DeSBI [KI-FOR 5363] (project ID: 459422098) and grant SELPHY-TS (project ID: 553038473).

References

- [1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.
- [2] Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL <https://openreview.net/forum?id=ePUVetPKu6>. Survey Certification, Expert Certification.
- [3] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013. URL <https://api.semanticscholar.org/CorpusID:32384865>.
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021.
- [5] Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. Objective drives the consistency of representational similarity across datasets. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=va3zmBXPat>.
- [6] Colin Conwell, Jacob S. Prince, Kendrick N. Kay, George A. Alvarez, and Talia Konkle. A large-scale examination of inductive biases shaping high-level visual representation in brains and machines. *Nature Communications*, 15(1):9383, Oct 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-53147-y. URL <https://doi.org/10.1038/s41467-024-53147-y>.
- [7] Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 2590–2607. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/11a7f429d75f9f8c6e9c630aeb6524b5-Paper-Conference.pdf.
- [8] Róbert Csordás, Christopher Potts, Christopher D Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations. In *The 7th BlackboxNLP Workshop*, 2024. URL <https://openreview.net/forum?id=NUQeYgg8x4>.

- [9] Ryan DeWolfe and Jeffery L. Andrews. Random models for fuzzy clustering similarity measures. *ArXiv*, abs/2312.10270, 2023. URL <https://api.semanticscholar.org/CorpusID:266348826>.
- [10] Diego Doimo, Aldo Glielmo, Alessio Ansuini, and Alessandro Laio. Hierarchical nucleation in deep neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, pages 7526–7536, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 9781713829546.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- [12] Amil Dravid, Yossi Gandelsman, Alexei A. Efros, and Assaf Shocher. Rosetta neurons: Mining the common units in a model zoo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1934–1943, October 2023.
- [13] Elena Facco, Maria d’Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140, Sep 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-11873-y. URL <https://doi.org/10.1038/s41598-017-11873-y>.
- [14] Thomas Fel, Victor Boutin, Louis Béthune, Remi Cadene, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 54805–54818. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/abf3682c9cf9245a0294a4bebe4544ff-Paper-Conference.pdf.
- [15] Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2711–2721, June 2023.
- [16] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. URL <https://arxiv.org/abs/2406.04093>.
- [17] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/77d2afcb31f6493e350fca61764efb9a-Paper.pdf.
- [18] Alvina Goh and Rene Vidal. Clustering and dimensionality reduction on Riemannian manifolds. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008. doi: 10.1109/CVPR.2008.4587422. ISSN: 1063-6919.
- [19] Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, Ramalingam Chellappa, Andrew Gordon Wilson, and Tom Goldstein. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *ArXiv*, abs/2310.19909, 2023. URL <https://api.semanticscholar.org/CorpusID:264818042>.
- [20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.

- [22] Moritz Herrmann, Daniyal Kazempour, Fabian Scheipl, and Peer Kröger. Enhancing cluster analysis via topological manifold learning. *Data Mining and Knowledge Discovery*, September 2023. ISSN 1573-756X. doi: 10.1007/s10618-023-00980-2. URL <https://doi.org/10.1007/s10618-023-00980-2>.
- [23] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=F76bwRSLeK>.
- [24] Eyke Hullermeier, Maria Rifqi, Sascha Henzgen, and Robin Senge. Comparing fuzzy partitions: A generalization of the rand index and related measures. *IEEE Transactions on Fuzzy Systems*, 20(3):546–556, 2012. doi: 10.1109/TFUZZ.2011.2179303.
- [25] Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. Resi: A comprehensive benchmark for representational similarity measures. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=PRvd03nfFi>.
- [26] Neehar Kondapaneni, Oisin Mac Aodha, and Pietro Perona. Representational similarity via interpretable visual concepts. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=ih3BJmIZbC>.
- [27] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2656–2666, 2018. URL <https://api.semanticscholar.org/CorpusID:43928547>.
- [28] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.
- [29] Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, abs/1802.03426, 2018. URL <https://api.semanticscholar.org/CorpusID:3641284>.
- [30] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), mar 2017. doi: 10.21105/joss.00205. URL <https://doi.org/10.21105/joss.00205>.
- [31] Davoud Moulavi, Pablo A. Jaskowiak, Ricardo J. G. B. Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *SDM*, 2014. URL <https://api.semanticscholar.org/CorpusID:16656312>.
- [32] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *The Journal of Machine Learning Research*, 21(1):184:7503–184:7542, January 2020. ISSN 1532-4435.
- [33] Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? *ArXiv*, abs/2305.00729, 2023. URL <https://api.semanticscholar.org/CorpusID:258426737>.
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- [35] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Neural Information Processing Systems*, 2021. URL <https://api.semanticscholar.org/CorpusID:237213700>.

- [36] Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803*, 2020.
- [37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL <https://doi.org/10.1007/s11263-015-0816-y>.
- [38] Shashank Shekhar, Florian Bordes, Pascal Vincent, and Ari Morcos. Objectives matter: Understanding the impact of self-supervised objectives on vision transformer representations. *ArXiv*, abs/2304.13089, 2023. URL <https://arxiv.org/abs/2304.13089>.
- [39] Ben Sorscher, Surya Ganguli, and Haim Sompolinsky. Neural representational geometry underlies few-shot concept learning. *Proceedings of the National Academy of Sciences*, 119(43):e2200800119, October 2022. doi: 10.1073/pnas.2200800119. URL <https://www.pnas.org/doi/10.1073/pnas.2200800119>.
- [40] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=4nPswr1KcP>.
- [41] Iliia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C. Love, Erin Grant, Jascha Achterberg, Joshua B. Tenenbaum, Katherine M. Collins, Katherine L. Hermann, Kerem Oktar, Klaus Greff, Martin N. Hebart, Nori Jacoby, Qiuyi Zhang, Raja Marjieh, Robert Geirhos, Sherol Chen, Simon Kornblith, Sunayana Rane, Talia Konkle, Thomas P. O’Connell, Thomas Unterthiner, Andrew Kyle Lampinen, Klaus-Robert Muller, Mariya Toneva, and Thomas L. Griffiths. Getting aligned on representational alignment. *ArXiv*, abs/2310.13018, 2023. URL <https://arxiv.org/2310.13018>.
- [42] Johanna Vielhaben, Stefan Bluecher, and Nils Strodthoff. Multi-dimensional concept discovery (MCD): A unifying framework with completeness guarantees. *Transactions on Machine Learning Research*, January 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=KxBQPz7HKh¬eId=QlqzbYGWI3>.
- [43] Sebastiano Vigna. A weighted correlation index for rankings with ties. In *Proceedings of the 24th International Conference on World Wide Web, WWW ’15*, page 1166–1176, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee. ISBN 9781450334693. doi: 10.1145/2736277.2741088. URL <https://doi.org/10.1145/2736277.2741088>.
- [44] Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7486–7496, 2022. URL <https://api.semanticscholar.org/CorpusID:254366577>.
- [45] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [46] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11682–11690, May 2021. doi: 10.1609/aaai.v35i13.17389. URL <https://ojs.aaai.org/index.php/AAAI/article/view/17389>.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We summarize the contribution of our work in the introduction and in the abstract.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We include a section on limitations of our work in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: We do not present any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We describe our methods, the dataset, models, metrics, and parameters we used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide our code and use models, datasets and methods that are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all models, dataset subset, methods and hyperparameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the standard error across layers for our sanity check in Tab. 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide details about the hardware we used and run-time for our experiments in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our work fully confirms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We briefly describe the positive impact through enhanced model understanding in the conclusion - we do not see any negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release new datasets or trained models, so there is no risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all models, datasets, and methods that we use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We provided our code and use and reference publicly available models, datasets and methods.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: We did not conduct studies with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: We did not conduct studies with human participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We did not use LLMs in the core method development of this work.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A HDBSCAN

After concept discovery with HDBSCAN, we compute concept proximity scores $\mathcal{P}_{\{\phi\}} = \{P(\phi_0), \dots, P(\phi_N)\}$, $P \in [0, 1]^n$ holds the concept membership scores $P^\alpha(\phi)$ of each concept C^α . These rely on the implementation of soft clustering with HDBSCAN from [30], which we formalize here for the reader's convenience.

Clustering HDBSCAN first transforms the feature space using a density-informed metric called *mutual reachability distance*

$$\text{MRD}(\phi_i, \phi_j) = \max(\text{coreDistance}_k(\phi_i), \text{coreDistance}_k(\phi_j), d(\phi_i, \phi_j)) \quad (4)$$

where $\text{coreDistance}_k(\phi)$ is the distance between a point ϕ and its k -nearest neighbor. Based on the mutual reachability distance between all pairs, a minimum spanning tree is constructed that connects all points and minimizes the sum of the edges weighted by MRD. From this, a hierarchical tree is constructed via robust single linkage clustering. The hierarchical tree is condensed by eliminating insignificant clusters and simplifying the hierarchy. This is achieved by selecting a range of *persistence* values λ , which are the inverses of the mutual reachability distances ($\lambda = 1/\text{MRD}$). Clusters that persist over significant ranges of λ , i.e. they are stable across multiple density levels, are retained, while clusters that exist only over narrow ranges of λ are considered noise and pruned from the tree. The result is a condensed tree that focuses on the most significant clusters. Finally clusters are extracted from the condensed tree either based on their stability across different density levels or simply the leaf nodes are identified as clusters.

Soft clustering with HDBSCAN The soft cluster membership scores combine a distance-based membership with and an outlier score.

For the distance-based membership to cluster C^α , first k exemplar points $\{\phi_i^\alpha\}$, $i \in [1, k]$, are extracted. A single centroid is not enough to characterize a cluster as its shape can be arbitrary. The exemplar points are the points within the leaf nodes beneath cluster C^α with maximum persistence λ in the condensed tree, i.e. the densest points where the cluster persists. Then, the distance membership score between a point ϕ and a cluster C^α is the inverse minimum distance across the exemplar points $\{\phi_i^\alpha\}$,

$$M^\alpha(\phi)_{\text{dist}} = \frac{[\min_i(d(\phi, \phi_i^\alpha))]^{-1}}{\sum_\beta [\min_j(d(\phi, \phi_j^\beta))]^{-1}}, \quad (5)$$

normalized across all clusters. The outlier-based membership compares a point's membership persistence to the total persistence of a cluster:

$$M^\alpha(\phi)_{\text{membership}} = \frac{\lambda_{\phi \rightarrow C^\alpha} - \lambda_{\text{birth}}^{C^\alpha}}{\lambda_{\text{max}}^{C^\alpha} - \lambda_{\text{birth}}^{C^\alpha}}. \quad (6)$$

Here, $\lambda_{\text{birth}}^{C^\alpha}$ is the persistence value at which cluster C^α first appears, i.e. its birth point in the condensed tree and $\lambda_{\phi \rightarrow C^\alpha}$ is the persistence value at which point ϕ would join cluster C^α . Finally, distance and outlier-based membership are combined with stronger emphasis on outlier-based membership,

$$M^\alpha(\phi) = (M^\alpha(\phi)_{\text{dist}})^{1/2} \cdot (M^\alpha(\phi)_{\text{membership}})^2, \quad (7)$$

and normalized $M_{\text{norm}}^\alpha(\phi) = M^\alpha(\phi) / \sum_\beta M^\beta(\phi)$. This membership score $M_{\text{norm}}^\alpha(\phi)$ can be interpreted as the probability that a point ϕ belongs to cluster C^α , given that the point belongs to some cluster,

$$M_{\text{norm}}^\alpha(\phi) \equiv P(\phi \in C^\alpha \mid \exists \beta : \phi \in C^\beta). \quad (8)$$

We want to compute the joint probability $P(\phi \in C^\alpha)$, which includes the probability that ϕ may be noise,

$$P(\phi \in C^\alpha) = P(\phi \in C^\alpha \mid \exists \beta : \phi \in C^\beta) P(\exists \beta : \phi \in C^\beta). \quad (9)$$

Here, $P(\exists \beta : \phi \in C^\beta)$ is the probability that ϕ belongs to some cluster. To estimate $P(\exists \beta : \phi \in C^\beta)$, the λ value at which ϕ would join the nearest cluster is compared to the maximum λ value of that cluster,

$$P(\exists \beta : \phi \in C^\beta) = \frac{\lambda_{\phi \rightarrow C^\alpha}}{\lambda_{\text{max}}^{C^\alpha}}, \quad (10)$$

where $\lambda_{\phi \rightarrow C^\alpha}$ is the persistence value at which point ϕ would join its nearest cluster C^α and $\lambda_{\max}^{C^\alpha}$ is the maximum λ value of cluster C^α . Thus, the final probability, that point ϕ belongs to cluster C^α is,

$$P^\alpha(\phi) = \frac{\lambda_{\phi \rightarrow C^\alpha}}{\lambda_{\max}^{C^\alpha}} \cdot M_{\text{norm}}^\alpha(\phi). \quad (11)$$

B Details on experimental setup

Here, we provide further details on the experiments.

ViT sources We list the URL of each Vision Transformer provided by the timm library [45]:

- FS: https://huggingface.co/timm/vit_base_patch16_224.augreg_in1k
- CLIP: https://huggingface.co/timm/vit_base_patch16_clip_224.openai
- DINO: https://huggingface.co/timm/vit_base_patch16_224.dino
- MAE: https://huggingface.co/timm/vit_base_patch16_224.mae

Hyperparameters for UMAP and HDBSCAN We tune hyperparameters of UMAP and HDBSCAN such that the density-based validity index DBCV is maximized across models and layers. Here, for DBCV, the average across clusters is weighted by their respective size such that the noise rate is indirectly included. We list the effect of the most influential hyperparameters that we tune and state the final value we used:

- **Minimal distance in UMAP:** a low minimal distance in UMAP enhances local cluster density but may also increase noise. We use a value of 0.01 in all experiments.
- **Number of neighbours in UMAP:** the number of neighbors controls the local structure, the smaller the finer it captures local neighborhoods but distorts global structure which is important for concept alignment analysis later. We use a value of 30 in all experiments.
- **Embedding dimensionality in UMAP:** We use the practical limit for HDBSCAN of $F' = 50$ in all experiments.
- **Minimum cluster size in HDBSCAN:** a too small minimum cluster size may identify noise as a cluster, whereas, when too large, distinct clusters will merge. We use a value of 50 in all experiments.
- **Min samples in HDBSCAN:** controls how conservative the algorithm is about noise. We need this to be rather low because of sampling limitations which means that most likely some concept manifolds are not sampled densely enough. We use a value of 20 in all experiments.

Additionally, we assume that clusters are rather uniform in size and select the leaf nodes in the HDBSCAN hierarchical condensed tree as clusters. Sampling one pooled SEQ token or one CLS token from each representation of images within a 25% subset of the ImageNet1-1k train set results in 315.770 feature vectors ϕ_i for clustering. We use the cuML [36] versions of HDBSCAN and UMAP for computation on the GPU.

In Tab. 2 and Tab. 3, we show a hyperparameter sensitivity study (at layer 11 of DINO SEQ representations) that varies UMAP and HDBSCAN parameters one at a time (the other hyperparameters are fixed at the default values stated above) and observe that concept discovery evaluation metrics are reasonably stable within a quite broad hyperparameter range. Only NRMSE shows a rather high range when varying the embedding dimension and UMAP minimal distance with minimum of < 1.0 for small embedding dimensions (≤ 40) and very small minimal distance. This is likely not a sign of a superior embedding, but rather an counter intuitive behavior of this metric due to extreme compression of pairwise distances in the embedding.

Cluster label in Concept Atlas To assign a label from the WordNet Hierarchy to each concept cluster, we first assign the ImageNet-1k label of the image from which a token is extracted to its representation feature vector ϕ_i . Then we map this to a label higher in the WordNet hierarchy by the mapping in Tab. 4. We then assign the most frequent label among the cluster members $\{\phi_j^\alpha\}$ to the cluster C^α .

Hyperparameter	Range	NRMSE	Noise ratio	DBCv	Robustness
Embedding dim. F'	20 – 100	0.7 < 9.5 < 15.1	0.66 < 0.66 < 0.66	0.56 < 0.58 < 0.59	0.84 < 0.84 < 0.87
UMAP min. dist.	0.005 – 0.5	0.7 < 9.5 < 13.6	0.64 < 0.66 < 0.72	0.53 < 0.58 < 0.58	0.81 < 0.85 < 0.86
UMAP neighbors	10 – 40	6.4 < 9.5	0.65 < 0.66 < 0.66	0.58 < 0.58 < 0.61	0.84 < 0.85 < 0.85

Table 2: Impact of UMAP hyperparameters on embedding and clustering quality at layer 11 of DINO SEQ representations. UMAP parameters are varied one at a time while all other hyperparameters are fixed at the default values stated above. Results are reported as min<default<max.

Hyperparameter	Range	Noise ratio	DBCv	Robustness
Min. cluster size	20 – 100	0.57 < 0.66 < 0.73	0.57 < 0.58 < 0.59	0.83 < 0.84 < 0.85
Min. samples	10 – 50	0.66 < 0.66 < 0.67	0.50 < 0.58 < 0.65	0.83 < 0.84 < 0.85

Table 3: Impact of HDBSCAN hyperparameters on clustering quality at layer 11 of DINO SEQ representations. HDBSCAN parameters are varied one at a time while all other hyperparameters are fixed at the default values stated above. Results are reported as min<default<max.

Computation of alignment Our concept-based alignment measure CBA is based on pairs of feature vectors (ϕ_i, ϕ_j) . To reduce run-time, we sub-sample 20% of the 315.770 feature vectors before computing CBA. Some concepts become under-sampled in this process, indicated by their total activation in the subsample being much lower than 20% of their activation over all samples. Such under-sampled concepts appear close to nearly all other concepts due to the missing activations. To address this, we exclude them by filtering out concepts whose total activation in the subsample is below 17% of their total activation across the entire dataset (this filters out around 4% of concepts in the examples shown here and in the main paper).

Runtime Collecting activations for a 25% subset of the ImageNet1-1k train set, concept discovery and concept discovery evaluation (NRMSE, DBCv) takes around 1.5 hours per layer. We performed around 30 runs for UMAP and HDBSCAN hyperparameter tuning. Computing scalar concept alignment for all pairs of layers depends on the total number of concepts, requiring about 1.25 hours for the FS model, which has the largest concept count. Concept-wise distance computations also scale with the number of concepts: for example, the fine-grained inter-model analysis shown in Fig. 6 in the main paper took about 6 hours on MAE vs. DINO at layer 10 with 477 and 684 concepts, respectively. All experiments were conducted on a Tesla V100 GPU.

C Baselines for concept discovery evaluation

Here, we describe two baselines for our evaluation of NLMCD concept discovery quality in Sec. 4.1 of the main paper. We present them alongside the concept discovery results for FS, CLIP, DINO, and MAE in Fig.7.

Random/ImageNet Baseline We feed ImageNet samples (the same subset as in Sec. 4.1) to a randomly initialized ViT. Although the input images themselves contain discriminative patterns, the model itself has no learned structure. Here, noise rate remains almost constant across layers, comparable to middle or later layers of trained models. UMAP preserves distances well in this setting, resulting in good NRMSE. DBCv scores (measuring cluster separation by density) are overall lower than in trained models but not as low as one might expect. Visually, clusters seem to capture pixel color similarities rather than semantic concepts (see examples in Fig.7). However, robustness of concept discovery is much lower: different initializations of UMAP and HDBSCAN lead to different color-driven clusters. In short, while input structure alone can induce clustering, it mainly reflects superficial color patterns rather than meaningful learned semantics with slightly reduced cluster separation and lower robustness.

Random/Noise Baseline Here, a randomly initialized ViT receives Gaussian noise patches. With no structure in the data or the model weights, HDBSCAN labels most points as noise, leading to a higher noise rate than concept discovery with ImageNet inputs or trained models. Similarly, DBCv values drop noticeably, whereas NRMSE remains only slightly higher, indicating that UMAP still preserves pairwise distances. Interestingly, robustness is relatively high in this scenario, simply because most tokens are consistently categorized as noise. This extreme setting underscores that

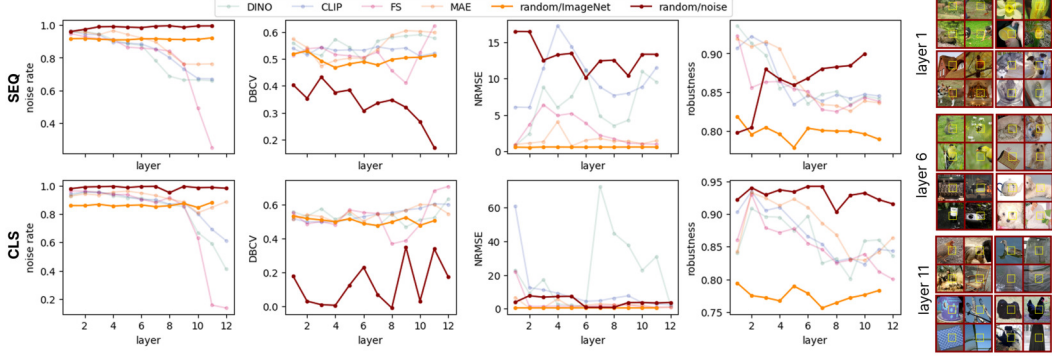


Figure 7: Two baselines for concept discovery evaluation based on a randomly initialized model with ImageNet or gaussian noise input. **Left:** The noise rate is the ratio of points classified as noise in HDBSCAN. DBCV is a density-based clustering validity index that contrasts intra- vs inter-cluster density with scores in $[-1, 1]$ where higher is better. NRMSE measures the root mean squared error between the distance matrix of the original and embedded activations, normalized by the average distance in the original embedding, and shows how faithfully the UMAP embedding captures the geometry of the representation. Robustness is measured between two runs by concept alignment from Eq. 2. Results are across layers for SEQ (upper row) and CLS (bottom row) token representations. **Right:** Randomly chosen examples for concepts discovered at layer 1, 6, 11 in the SEQ representations of the randomly initialized model, all corresponding to color patterns.

both structured data and learned parameters are essential for creating meaningful clusters in concept discovery.

Summary Overall, the Random/ImageNet baseline shows that the input distribution alone can drive clustering based on superficial pixel color regularities in representations, but naturally fails to produce semantically rich clusters driven by the model’s learned concepts. The Random/Noise baseline confirms that without both learned parameters and structured data, clustering breaks down almost entirely, highlighting the fundamental role of learned representations for robust and meaningful concept discovery.

D Concept Formation Graphs

Notation. Let $C^{l,\alpha}$, with $\alpha \in 1, \dots, N^l$, denote a concept cluster in layer l . For a token $i \in 1, \dots, k$, we use the notation $\phi_i^l \in C^{l,\alpha}$ to indicate that the token representation ϕ_i^l in layer l is assigned to that cluster. Let $C^{l,*}$ denote the target concept for which a concept formation graph (CFG) is constructed.

Given k tokens sampled from the training dataset and their cluster assignments, the algorithm for constructing the CFG is defined as follows:

1. **Transition matrix calculation:** First, we compute transition matrices $T_{l,l+1} \in \mathbb{Z}^{N^l \times N^{l+1}}$ for each pair of consecutive layers $(l, l+1)$. Each entry represents the count of tokens transitioning from a concept $C^{l,\alpha}$ in layer l to a concept $C^{l+1,\beta}$ in layer $l+1$:

$$(T_{l,l+1})_{\alpha\beta} = \#\{i \in \{1, \dots, k\} : \phi_i^l \in C^{l,\alpha} \text{ and } \phi_i^{l+1} \in C^{l+1,\beta}\} \quad (12)$$

where $\#\{\cdot\}$ denotes the count of tokens.

2. **Recursive graph construction:** Initializing the set of CFG nodes with the target concept node $C^{l,*}$, we recursively add all predecessor concept nodes whose “contribution” (the proportion of incoming transitions) surpasses a specified threshold τ . Formally, suppose a concept $C^{l+1,\beta}$ in layer $l+1$ has been added to the CFG. Then, for each concept $C^{l,\alpha}$ in

layer l , we include the edge $(C^{l,\alpha}, C^{l+1,\beta})$ and node $C^{l,\alpha}$ to the CFG if:

$$\frac{(T_{l,l+1})_{\alpha\beta}}{\sum_{\gamma=1}^{N^l} (T_{l,l+1})_{\gamma\beta}} > \tau. \quad (13)$$

The resulting CFG is a binary, unidirectional graph. Fig. 9, 10, and 8 illustrate additional exemplary CFGs for CLIP and DINO. The CFGs were constructed using the same $k = 315,770$ tokens from the ImageNet training dataset that were used for concept discovery, with the threshold parameter set to $\tau = 0.05$. In each image, the “concept” token is highlighted in yellow. The concepts in Fig. 3 of the main text are human-labeled.

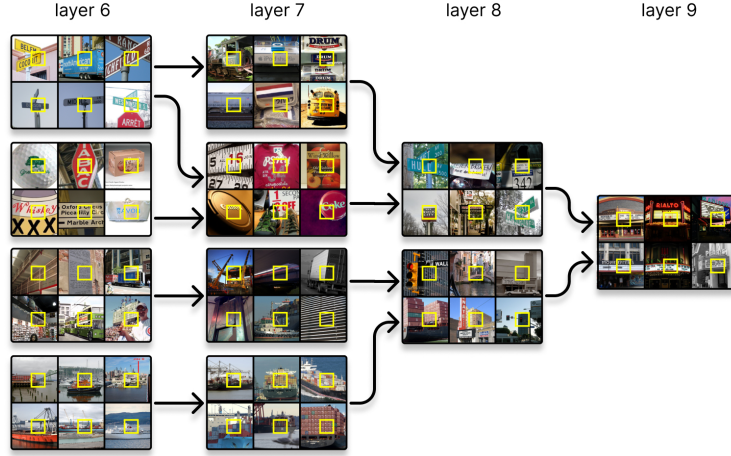


Figure 8: Concept formation graph for a concept in layer 9 of DINO. Each concept is represented by six randomly sampled images containing a token assigned to that concept (highlighted in a yellow frame).

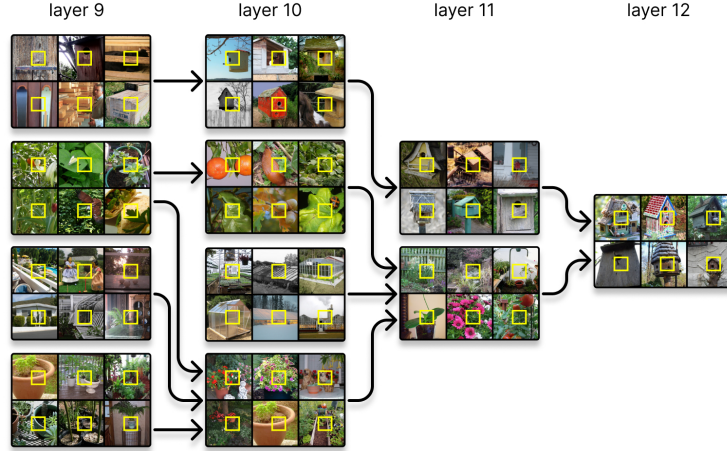


Figure 9: Concept formation graph for a concept in layer 12 of CLIP. Each concept is represented by six randomly sampled images containing a token assigned to that concept (highlighted in a yellow frame).

E Qualitative comparison against linear and spherical concept discovery

Figure 11 illustrates the concept atlases for four concept discovery methods (PCA, MCD, KMeans, and NLMCD) in the concept alignment sanity check of Sec. 4.2 in the main paper. For a direct qualitative comparison, we also show concept atlases at layer six for CLIP SEQ token concepts using

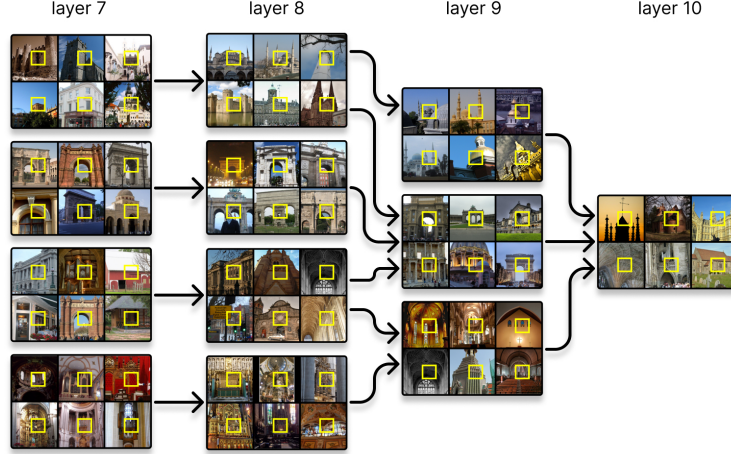


Figure 10: Concept formation graph for a concept in layer 10 of CLIP. Each concept is represented by six randomly sampled images containing a token assigned to that concept (highlighted in a yellow frame).

each method (Fig. 12–15). We segment each atlas into 15 groups via KMeans (indicated by color) and display random examples of concepts within each group.

Visually, MCD, KMeans, and NLMCD exhibit more coherent concept clusters than PCA. Moreover, when examining groups of close concepts in each atlas, we find that NLMCD and KMeans yield more structured organization and greater semantic consistency within concept groups than MCD or PCA (e.g., note how the bright yellow cluster in the MCD atlas mixes sundown/horizon, fur, and shiny object concepts). While the difference in semantic consistency is subtle between KMeans and NLMCD in this example, the results in Sec. 4.2 show that KMeans is on par with NLMCD in half of the test settings but never outperforms it. Nevertheless, KMeans can still serve as a slightly less accurate but more computationally cheap alternative to NLMCD.

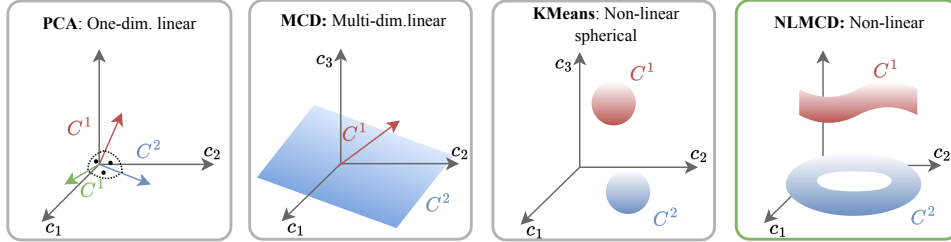


Figure 11: Illustration of the four different concept definitions that are compared for measuring concept alignment in Sec. 4.2 in the main paper (in three dimensions).

F Concept alignment analysis

F.1 CLS representations

We investigate concept-based alignment within and across models based on the CLS token representations analogous to the SEQ token analysis in the main paper.

Intra-model alignment We first compare the intra-model alignment heatmaps for CLS representations across layers (top row of Fig. 16), measured by CBA, with the same analysis for SEQ tokens presented in the main paper. For the FS model, the CLS representations exhibit a pattern very similar to that of the SEQ representations. Likewise, CLIP and MAE show largely comparable CLS–SEQ alignment profiles, but their earliest layers display noticeably lower alignment for CLS,



Figure 12: Detailed concept atlas for PCA SEQ token concepts at layer six of CLIP. This is based on a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this concept atlas corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 15 groups (indicated by color) and show four random concepts for each group.

suggesting that the model might rely less on the CLS token in early layers. With respect to class label alignment, intrinsic dimensionality, and the number of discovered concepts, the CLS patterns in Fig. 17 also broadly resemble the SEQ results, except for DINO. In DINO’s CLS tokens, alignment across layers is substantially lower than in the SEQ tokens, and the concept atlas in layer 11 appears less semantically organized. One plausible explanation is that the UMAP embeddings for DINO’s CLS tokens (layers 7–11) exhibit higher NRMSE, indicating reduced faithfulness of the embedding to the original activations, and thereby less faithful concepts.

Inter-model alignment Second, we analyze how the CLS representations between two different models differ and present CBA alignment between all layers and model in the upper part of 18. Like for the SEQ representations, CLS representations at layers of the first are more aligned than those of the second half across all models pairs, suggesting that basic foundational features are learned similarly across models, while later layers diverge as the models specialize to concepts serving their pre-training task. However, the overall alignment between models is weaker for CLS representations than for SEQ, also in low layers. Next, we zoom in into the distance $d_{cross}(P^\alpha, P^\beta)$ between concept pairs from CLS representations of MAE layer 11 and CLIP layer 11 in the right part of 18. Next to the pairwise distance matrix $d_{cross}(P^\alpha, Q^\beta)$, we present the distribution of the distances between nearest matches, and randomly chosen examples of concepts from MAE and their nearest matches in CLIP. Similar to the comparison of SEQ concepts between MAE and DINO in Sec. 4.3 in the main paper, we find that 80% of CLS concepts in MAE are similar to only 5 *singular* concepts in CLIP. Lastly, among the randomly chosen examples, visual discrepancy is more pronounced for high-distance concepts than for the other pairs with lower distance.

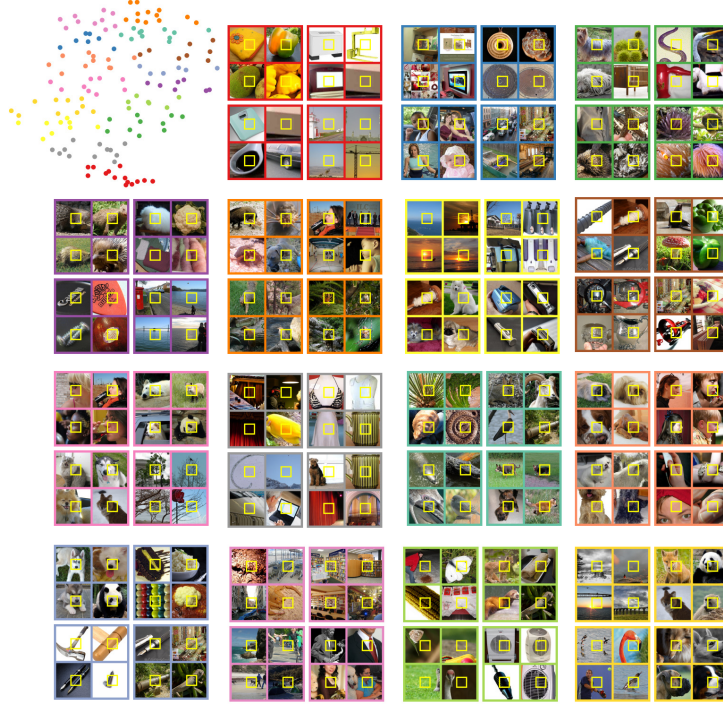


Figure 13: Detailed concept atlas for MCD SEQ token concepts at layer six of CLIP. Detailed concept atlas for PCA SEQ token concepts at layer six of CLIP. This is based on a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this concept atlas corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 15 groups (indicated by color) and show four random concepts for each group.

F.2 Additional results for SEQ representations

Intra-model To give a more detailed view of the organization of concepts across the layers of one model, we select the DINO model and show the respective concept atlases at layer one, six and eleven in 19, 20, and 21, respectively. To give an overview of the structure within a concept atlas, we group the concepts in the UMAP embedding via KMeans and show four random concepts for each group. In layer one, many concepts correspond to color, in layer six, they represent mostly textures, and in layer eleven they correspond to abstract concepts. Mostly, concepts within a group are of similar nature.

Inter-model In the main paper, we show fine-grained inter-model concept distances between DINO and MAE at layer ten. Here, we add fine-grained concept distance analysis between DINO and MAE, as well as between MAE and CLIP in 22. Again, we show the full pairwise distance matrix as well as how distances between closest matching pairs are distributed. Similar to the comparison of SEQ concepts between MAE and DINO in Sec. 4.3 in the main paper, we find that in the case of DINO vs. CLIP, 65% of CLIP concepts most closely match with only 2 *singular* concepts in DINO; for MAE vs. CLIP, 86% of MAE concepts match with 4 concepts in CLIP.

Now, we examine differences in concept alignment between pairs of layers. Specifically, we compare layer 10 in MAE/DINO (see Fig. 6) and the same layer in CLIP/DINO (see Fig. 16), which exhibit the same overall concept alignment (0.74). To summarize fine-grained differences in concept-alignment, we group concept nearest-neighbor-pairs into the WordNet categories from Fig. 4 (88% of concept pairs are in the same category for MAE/DINO, 93% for CLIP/DINO). We then compute the mean concept distance within each concept category and find that 1) canine and insect concepts have high distance in both model pairs, while equipment and material concept have low distance in both pairs; 2) fish concepts are similar in MAE/DINO but have high distance in CLIP/DINO, while sport concepts



Figure 14: Detailed concept atlas for KMeans SEQ token concepts at layer six of CLIP. Detailed concept atlas for PCA SEQ token concepts at layer six of CLIP. This is based on a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this concept atlas corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 15 groups (indicated by color) and show four random concepts for each group.

show the opposite trend. This exemplifies a scenario where two representations are similarly aligned by the overall score, but fine-grained concept alignment differs.



Figure 15: Detailed concept atlas for NLMCD SEQ token concepts at layer six of CLIP. Detailed concept atlas for PCA SEQ token concepts at layer six of CLIP. This is based on a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this concept atlas corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 15 groups (indicated by color) and show four random concepts for each group.

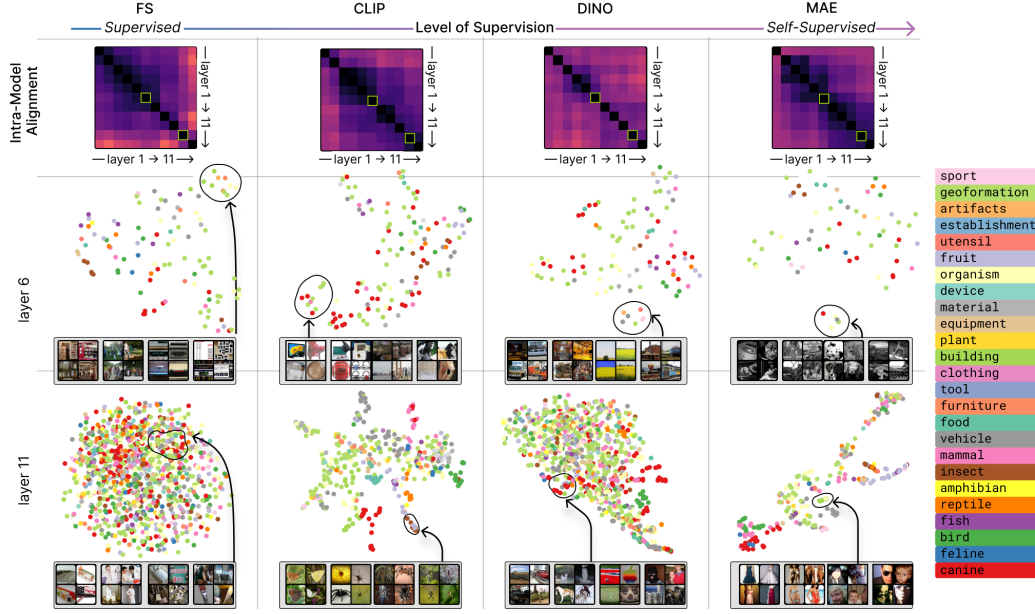


Figure 16: Intra-model relationships based on CLS representations across layers. In the **upper row**, we show CBA to visualize how representations are transformed across layers of the models (darker pixels correspond to higher alignment). In the **center and bottom row** we zoom into the representations at layer 6 and 11 of each model and partition the scalar CBA alignment into single concepts. We show a UMAP embedding constructed from the pairwise distance of concept measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this *concept atlas* corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster C^α marked by a yellow box in the entire image.

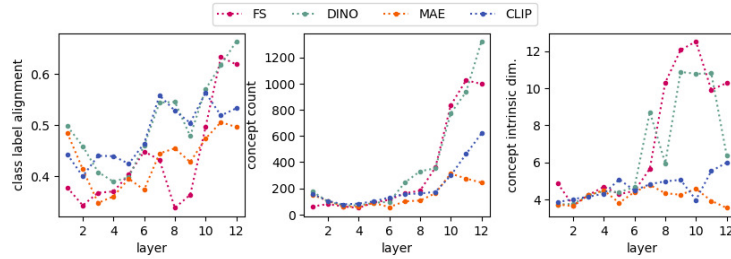


Figure 17: Class label alignment (based on CBA), concept count, and the average intrinsic dimensionality (based on [13]) across concepts for CLS representations supplement the intra-model alignment analysis, by providing insights into how well the model aligns with ImageNet-1k labels, the spatial organization of tokens, and the complexity of the learned concepts as they evolve through the layers.

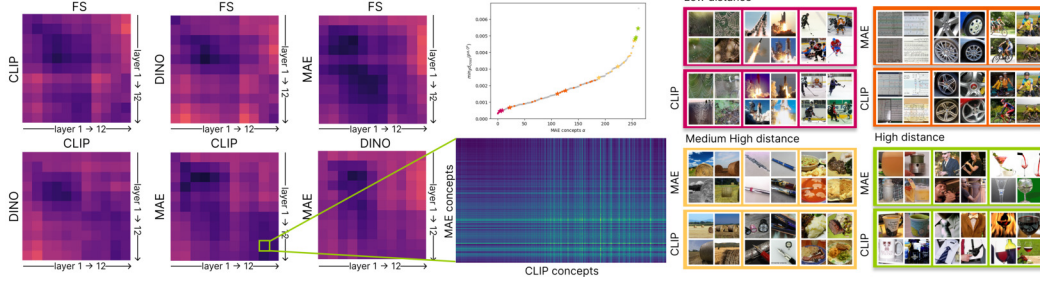


Figure 18: Inter-model relationships based on CLS representations across layers. **Right:** We show CBA to visualize how representations differ between the models (darker pixels correspond to higher alignment). **Left:** We zoom in into the concept-wise distances $d_{cross}(P^\alpha, P^\beta)$ between the representation of in MAE and CLIP at layer 11. Alongside the pairwise distance matrix $d_{cross}(P^\alpha, Q^\beta)$, we present randomly chosen examples of concepts from MAE and their nearest matches in CLIP, and the distribution of the distances between nearest matches. We illustrate some random example pairs with low to high distance (marked by a star in the distance distribution plot).



Figure 19: We zoom into the SEQ representations at DINO layer 1 and show a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this *concept atlas* corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 7 groups and show four random concepts for each group. Concepts representing similar colors lie within the same group, e.g. shades of blue in the blue group or red and orange in the red group.

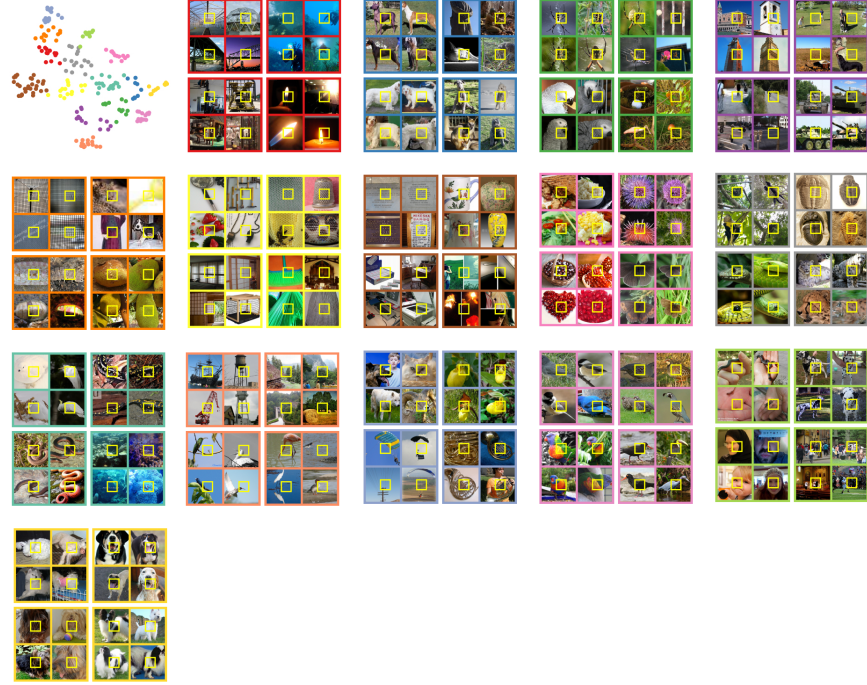


Figure 20: We zoom into the SEQ representations at DINO layer 6 and show a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this *concept atlas* corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 15 groups and show four random concepts for each group. Most concepts represent a pattern or texture which are similar within each group.

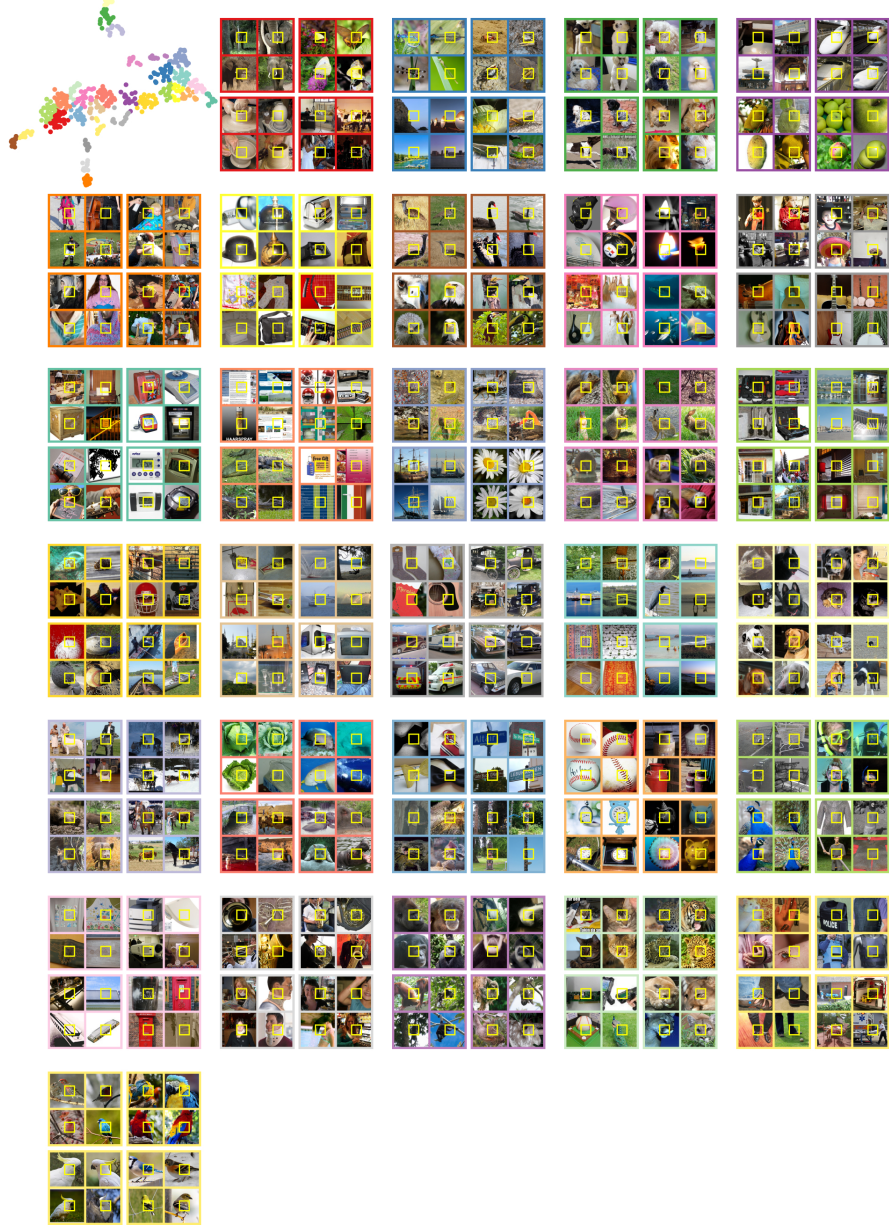


Figure 21: We zoom into the SEQ representations at DINO layer 11 and show a UMAP embedding constructed from the pairwise distance of concepts measured by $d_{cross}(P^\alpha, P^\beta)$. Each point in this *concept atlas* corresponds to a distinct concept P^α . To convey their meaning, we show four random input tokens from the members of the concept cluster P^α . We dissect the concept atlas into 30 groups and show four random concepts for each group. For most groups, these are semantically similar.

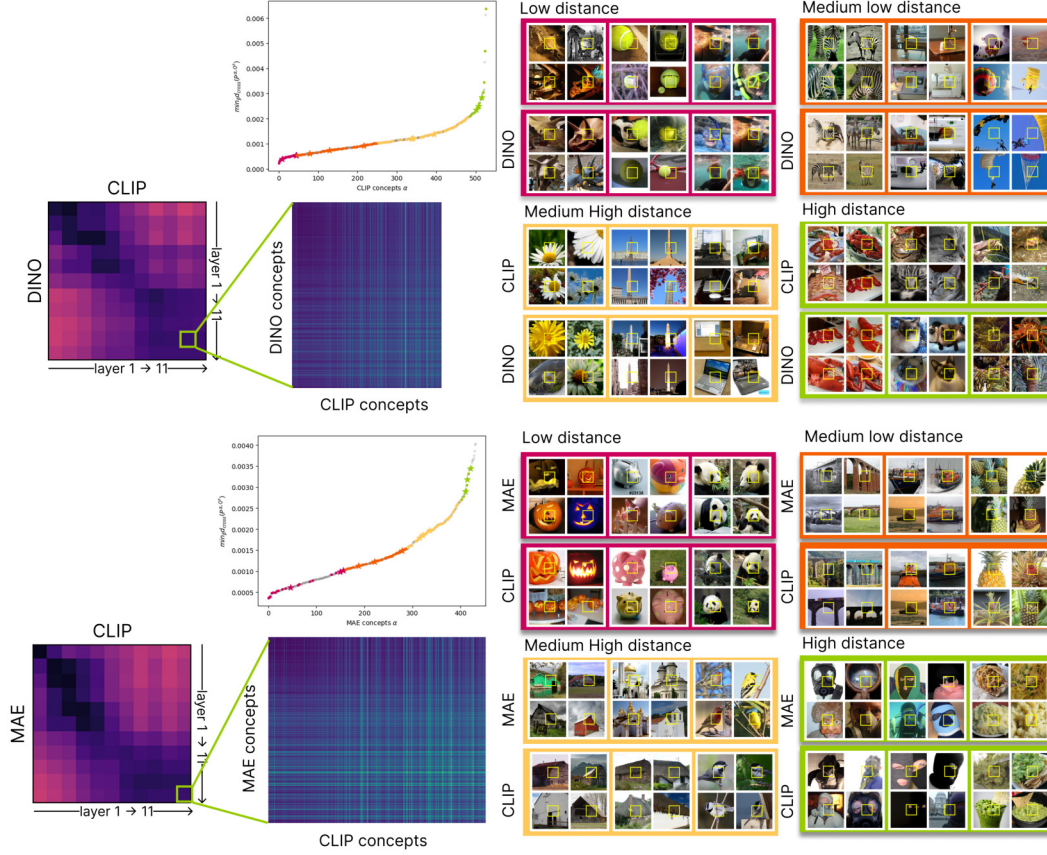


Figure 22: CBA of SEQ concepts across layers of CLIP and DINO, as well as CLIP and MAE (darker pixels correspond to higher alignment). We zoom in into the concept-wise distances $d_{cross}(P^\alpha, P^\beta)$ between the representation of layer ten in CLIP and DINO, as well as layer 11 in CLIP and MAE. Alongside the pairwise distance matrix $d_{cross}(P^\alpha, Q^\beta)$, we present randomly chosen examples of concepts from MAE and their nearest matches in DINO/CLIP, and the distribution of the distances between nearest matches. We illustrate some random example pairs with low to high distance (marked by a star in the distance distribution plots).

Category	ImageNet-1k class
amphibian	European fire salamander axolotl bullfrog common newt eft spotted salamander tailed frog tree frog
artifacts	Afghan hound Band Aid Dutch oven Petri dish abacus ashean backpack ballpoint bannister barrel bath towel bathtub beacon beaker beer bottle beer glass bell cote binder birdhouse book jacket bottlecap brass breakwater breastplate broom bucket cannon carousel carton cassette chain mail chainlink fence chiffonier cleaver cliff dwelling cloak clog cocktail shaker coffee mug comic book cowboy boot crate crib crutch cuirass cup diaper dishwasher dock envelope espresso maker face powder fig fire screen flagpole fountain fountain pen gasmask goblet grasshopper grille grocery store guillotine hair spray hand blower holster honeycomb iron "jack-o-lantern" joystick ladle lampshade lens cap library lipstick lotion mailbag mailbox manhole cover mask matchstick maze measuring cup megalith menu microwave minibus mixing bowl mobile home mortar mortarboard mosquito net mountain tent muzzle necklace obelisk packet paddle patio pedestal pencil box pencil sharpener perfume pickelhaube picket fence pier piggy bank pill bottle pillow pitcher plastic bag plate rack pole pop bottle pot prayer rug purse quill quilt racket radio rain barrel refrigerator rotisserie rubber eraser running shoe safe saltshaker scabbard school bus schooner scoreboard shield shoji shopping basket shower curtain ski mask sleeping bag sliding door soap dispenser soup bowl space bar spotlight steel arch bridge stone wall stove street sign stretcher sunscreen suspension bridge swab swing teddy television thatch theater curtain thimble tile roof totem pole traffic light tray triumphal arch trolleybus tub turnstile umbrella vacuum vase viaduct waffle iron washbasin washer water bottle water jug water tower web site whiskey jug window screen window shade wine bottle worm fence wreck yurt
bird	African grey American coot American egret European gallinule albatross bald eagle bee eater bittern black grouse black stork black swan brambling bulbul bustard chickadee cock coucal crane dowitcher drake flamingo goldfinch goose great grey owl hen hornbill house finch hummingbird indigo bunting jacamar jay junco king penguin kite limpkin little blue heron lorikeet macaw magpie ostrich oystercatcher partridge pelican prairie chicken ptarmigan quail red-backed sandpiper red-breasted merganser redshank robin ruddy turnstone ruffed grouse spoonbill sulphur-crested cockatoo toucan vulture water ouzel white stork
building	apiary barn boathouse castle church cinema greenhouse home theater monastery mosque palace planetarium prison restaurant stage stupa vault
canine	African hunting dog Airedale American Staffordshire terrier Appenzeller Arctic fox Australian terrier Bedlington terrier Bernese mountain dog Blenheim spaniel Border collie Border terrier Boston bull Bouvier des Flandres Brabancon griffon Brittany spaniel Cardigan Chesapeake Bay retriever Chihuahua Dandie Dinmont Doberman English foxhound English setter English springer EntleBucher Eskimo dog French bulldog German short-haired pointer Gordon setter Great Dane Great Pyrenees Greater Swiss Mountain dog Ibizan hound Irish setter Irish terrier Irish water spaniel Irish wolfhound Italian greyhound Japanese spaniel Kerry blue terrier Labrador retriever Lakeland terrier Leonberg Lhasa Maltese dog Mexican hairless Newfoundland Norfolk terrier Norwegian elkhound Norwich terrier Pekinese Pembroke Pomeranian Rhodesian ridgeback Rottweiler Saint Bernard Saluki Samoyed Scotch terrier Scottish deerhound Sealyham terrier Shetland sheepdog Shih-Tzu Siberian husky Staffordshire bullterrier Sussex spaniel Tibetan mastiff Tibetan terrier Walker hound Weimaraner Welsh springer spaniel West Highland white terrier Yorkshire terrier affenpinscher basenji basset beagle black-and-tan coonhound bloodhound bluetick borzoi briard bull mastiff cairn chow clumber cocker spaniel collie coyote curly-coated retriever dalmatian dhole dingo flat-coated retriever giant schnauzer golden retriever grey fox groenendael hyena keeshond kelpie kit fox komondor kuvasz malamute malinois miniature pinscher miniature poodle miniature schнауzer otterhound papillon pug red fox red wolf redbone schipperke silky terrier soft-coated wheaten terrier standard poodle standard schnauzer timber wolf toy poodle toy terrier vizsla whippet white wolf wire-haired fox terrier Christmas stocking Loafer Old English sheepdog Windsor tie abaya academic gown apron bathing cap bearskin bib bikini bolo tie bonnet bow tie brassiere bulletproof vest cardigan chest cowboy hat crash helmet dishrag feather boa fur coat gown handkerchief hook hoopskirt jean jersey kimono knee pad lab coat maillot military uniform miniskirt mitten overskirt pajama paper towel poncho sandal sarong seat belt shower cap sock sombrero stole suit sweatshirt swimming trunks trench coat velvet vestment wallet wig wool
device	accordion acoustic guitar analog clock assault rifle banjo barometer bassoon binoculars bow buckle bullet train candle car mirror car wheel cash machine cello chime combination lock desktop computer digital clock digital watch disk brake drum drumstick electric fan electric guitar flute gas pump gong hair slide hammer hamper hand-held computer hard disc harmonica harp hatchet horn hourglass laptop lighter loudspeaker loupe magnetic compass maraca marimba maypole microphone missile monitor mouse mousetrap neck brace notebook oboe odometer oil filter organ oxygen mask paddlewheel padlock paintbrush panpipe parking meter pick "potters wheel" projector puck radiator radio telescope remote control revolver rifle safety pin sax scale screen sewing machine ski slide rule slot slug snorkel solar dish space heater spider web steel drum stethoscope stopwatch strainer sundial sunglass sunglasses switch syringe thresher toaster torch tripod trombone typewriter keyboard upright vending machine violin wall clock whistle
equipment	CD player Polaroid camera balance beam barbell "carpenters kit" cassette player cellular telephone computer keyboard croquet ball crossword puzzle dial telephone drilling platform dumbbell golf ball golfcart horizontal bar iPod jigsaw puzzle modem oscilloscope parachute parallel bars pay-phone photocopier ping-pong ball plate punching bag reel reflex camera soccer ball tape player
establishment	bakery barbershop bookshop butcher shop confectionery shoe shop tobacco shop toyshop
feline	Egyptian cat Persian cat Siamese cat catamount cheetah coil cougar leopard lion panther snow leopard tabby tiger tiger cat
fish	anemone fish barracouta coho eel electric ray gar goldfish great white shark hammerhead lionfish puffer rock beauty stingray sturgeon tench tiger shark
food	Christmas loaf bagel burrito carbonara cheeseburger chocolate sauce consommé cucumber dough eggnog espresso guacamole hay hot pot hotdog ice cream ice lolly mashed potato meat loaf pizza potpie pretzel red wine trifle
fruit	Granny Smith acorn buckeye hip jackfruit lemon orange pineapple rapeseed strawberry
furniture	altar barber chair bassinet beaver bookcase china cabinet cradle desk dining table entertainment center file folding chair four-poster medicine chest milk can mink otter park bench pool table rocking chair studio couch table lamp throne toilet seat wardrobe
geological formation	alp bubble cliff coral reef dome geyser lakeside promontory sandbar seashore valley volcano
insect	ant bee cabbage butterfly cicada cricket damselfly dragonfly dung beetle fly ground beetle lacewing ladybug leaf beetle leafhopper long-horned beetle lycanid mantis monarch peacock rhinoceros beetle ringlet sulphur butterfly tiger beetle walking stick weevil
mammal	African elephant American black bear Angora Arabian camel Indian elephant Madagascar cat Sus scrofa armadillo baboon bighorn bison black-footed ferret brown bear capuchin chimpanzee colobus dugong echidna fitch fox squirrel gazelle gibbon gorilla grey whale guenon guinea pig hamster hare hartebeest hippopotamus ibex ice bear impala indri killer whale koala langur lesser panda llama macaque marmoset marmot meerkat mongoose orangutan ox panda patas platypus polecat porcupine proboscis monkey ram sea lion siamang sloth bear spider monkey squirrel monkey three-toed sloth titi tusker wallaby warthog water buffalo wombat wood rabbit zebra
material	chain cornet doormat groom knot spindle toilet tissue
musical	grand piano
organism	American lobster Dungeness crab German shepherd admiral agaric badger ballplayer barn spider black and gold garden spider black widow bolete boxer brain coral centipede chiton cockroach conch coral fungus crawfish dam ear earthstar fiddler crab flatworm garden spider gyromitra harvester harvestman hen-of-the-woods hermit crab hog howler monkey isopod jellyfish king crab mushroom nematode nipple printer rock crab rule scorpion scuba diver sea cucumber sea slug sea urchin snail spiny lobster starfish stinkhorn tarantula tick trilobite weasel wing wolf spider
plant	acorn squash artichoke banana bell pepper broccoli butternut squash cardoon cauliflower corn custard apple daisy head cabbage ocarina pinwheel pomegranate sea anemone sorrel spaghetti squash "yellow ladys slipper" zucchini
reptile	African chameleon African crocodile American alligator American chameleon Gila monster Indian cobra Komodo dragon agama alligator lizard banded gecko boa constrictor box turtle common iguana diamondback frilled lizard grass snake green lizard green mamba green snake hognose snake king snake leatherback turtle loggerhead mud turtle night snake ringneck snake rock python sand viper sea snake sidewinder terrapin thunder snake triceratops vine snake water snake whiptail
sport	baseball basketball football helmet rugby ball tennis ball volleyball
tool	can opener chain saw corkscrew lawn mower letter opener lumbermill nail plane plow plunger power drill screw screwdriver shovel
utensil	Crock Pot caldron coffeepot frying pan spatula teapot wok wooden spoon
vehicle	Model T aircraft carrier airliner airship ambulance amphibian balloon barrow beach wagon bicycle-built-for-two bobsled cab canoe catamaran chambered nautilus container ship convertible dogsled electric locomotive fire engine fireboat forklift freight car garbage truck go-kart gondola half track horse cart jeep jinrikisha lifeboat limousine liner minivan moped motor scooter mountain bike moving van oxcart passenger car pickup pirate racer recreational vehicle shopping cart snowmobile snowplow space shuttle speedboat sports car steam locomotive streetcar submarine tank tow truck tractor trailer truck tricycle trimaran unicycle wagon warplane yawl

Table 4: Mapping between categories from the WordNet Hierarchy and the ImageNet-1k classes used for assigning a category to the concept clusters.