VIT ATTENTION WEIGHT AS A HIERARCHICAL TREE

Anonymous authors
Paper under double-blind review

ABSTRACT

While attention weights in Vision Transformers (ViTs) have become a central signal for tasks such as object localization, segmentation, and attribution, they still remain noisy and difficult to interpret. To address this, we analyze attention at the level of individual query-key pairs rather than aggregated attention weights. By defining the concatenation of query and key vectors as a *QK feature*, we first show that QK features exhibit structured patterns rather than forming a single homogeneous cluster. Based on this observation, we organize QK features into a hierarchical structure via recursive clustering, yielding a *QK tree* for each attention head. We verify that each node in the QK tree encodes a consistent semantic role, showing that distinct QK feature patterns capture distinct semantics. To interpret these roles, we introduce the *role card*, a diagnostic tool that visualizes the functional behavior of each node. Using this decomposition, we further identify *position-indicator nodes*, which show that spatial selectivity is localized within specific nodes rather than distributed across the entire head. Finally, we demonstrate that selecting a single *object-centric node* yields a simple and effective object localization method.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have become a default architecture across a wide range of vision tasks, including image classification, object detection (Carion et al., 2020), semantic segmentation (Strudel et al., 2021), and image generation (Peebles & Xie, 2023). A defining feature of ViTs is the attention mechanism, whose weights explicitly capture information flow between tokens. As a result, attention weights have become a primary signal for a wide range of downstream tasks, including training-free object discovery and saliency that exploit self-supervised attention maps (Caron et al., 2021; Siméoni et al., 2021), weakly supervised semantic segmentation (Xu et al., 2022a;b), multiple-instance learning (Ilse et al., 2018), and spatial control and attribution in text-to-image diffusion models (Chefer et al., 2023; Tian et al., 2024).

Yet attention maps remain noisy and difficult to interpret. A non-trivial fraction of mass concentrates on attention-sink tokens that carry little task signal (Xiao et al., 2024; Darcet et al., 2024; Kang et al., 2025); raw weights are often not faithful explanations of model behavior (Jain & Wallace, 2019; Kobayashi et al., 2020); and a sizeable subset of heads can be pruned at no functional cost (Voita et al., 2019). A line of work has therefore sought more reliable attention-based explanations, either by aggregating signals across layers (Abnar & Zuidema, 2020; Chefer et al., 2021a;b) or by stepping outside attention to pooling- and gradient-based attribution (Zhou et al., 2016; Selvaraju et al., 2017; Wang et al., 2024). Even with these refinements, downstream pipelines built directly on attention often transfer poorly across backbones (Darcet et al., 2024).

We argue that this brittleness stems from analyzing attention at a coarse level. An attention weight is a softmax over a single scalar dot product (*i.e.*, $\mathbf{q}_i^{\ell,h} \cdot \mathbf{k}_j^{\ell,h}$), meaning that many distinct query-key patterns can yield the same value while playing qualitatively different roles (Pan et al., 2024; Doshi et al., 2026). This degeneracy implies that an attention map collapses heterogeneous interactions into a single scalar field, potentially obscuring meaningful structure. We therefore step back from aggregated attention and instead analyze the raw query and key representations directly.

Defining the concatenation $[\mathbf{q}_i^{\ell,h}; \mathbf{k}_j^{\ell,h}]$ as a *QK feature* (Figure 1a), we first show that its empirical distribution exhibits structured patterns rather than forming a single homogeneous cluster. Based on this observation, we organize QK features into a hierarchical structure via recursive clustering (Steinbach, 2000), yielding a *QK tree* (Figure 1b) for each attention head. Each node in the QK tree therefore

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

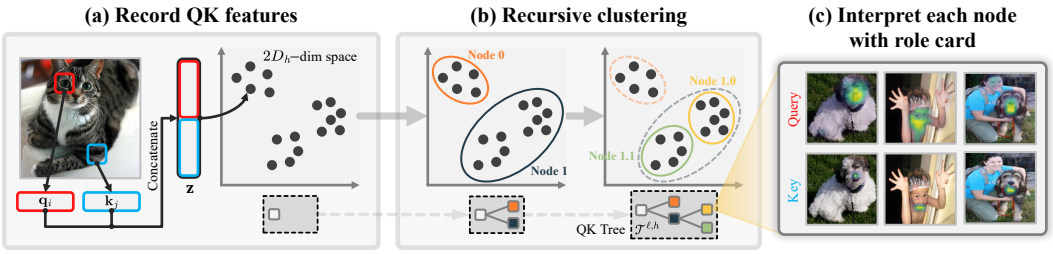


Figure 1: Overview of our pipeline. Given an attention head, (a) we simply concatenate query and key vectors, (b) build a QK tree via recursive clustering, (c) and interpret each node with a role card.

represents a distinct pattern of QK features. We further show that the queries and keys assigned to each node consistently fall on semantically coherent regions across images, indicating that each pattern carries a consistent semantic role. Taken together, these findings suggest that attention heads internally decompose into a hierarchy of functionally distinct sub-patterns.

Equipped with this decomposition, we introduce the *role card* (Figure 1c), a diagnostic tool that characterizes the functional role of each node by visualizing its associated QK feature cluster. Through role cards, we visualize the distinct functional roles captured by different nodes within an attention head and show that nodes at greater depths capture more specific semantic roles. Moreover, we identify *position-indicator nodes*, whose associated query-key pairs consistently activate only when the query and key occupy specific spatial regions within an image. These findings reveal that spatial selectivity in attention is not uniformly distributed across a head, but is instead localized to specific nodes within the hierarchy.

Finally, we show that the QK tree decomposition is not only interpretable but also actionable. An attention head mixes multiple QK patterns, and only a subset of its nodes carries object-relevant information. By simply identifying and selecting these nodes, we obtain an object localization method that outperforms both single-head attention maps and layer-level aggregations of attention heads. This result suggests that attention heads may conceal useful internal structure that becomes visible at finer levels of analysis.

Contribution. We make the following discoveries: ① QK features form structured, non-unimodal distributions, ② Distinct QK patterns are organized into hierarchical QK trees via recursive clustering, ③ QK tree nodes, each representing a distinct QK feature pattern, capture consistent semantic roles, ④ Role cards provide a diagnostic tool for interpreting the functional role of each node, ⑤ Selecting an object-centric node improves localization over head- and layer-level attention.

2 Related Work

Head-level analyses in vision transformers. Several works have analyzed the functional roles of attention heads in ViTs. Caron et al. (2021) highlight the emergent foreground localization behavior of `[CLS]`-to-patch attention in DINO, while Walmer et al. (2023) compare supervised, contrastive, and reconstructive ViTs and identify classes of attention heads (including offset-local heads) whose roles depend on the training objective. Gandelsman et al. (2024) decompose CLIP's image representation as a sum over heads and assign property-specific roles (location, shape, etc.) by projecting each head's contribution into the text embedding space. Balasubramanian et al. (2024) extend this text-based decomposition framework beyond CLIP to DeiT, DINO, DINOv2, Swin, and MaxViT. Kang et al. (2025) similarly find that a small number of attention heads explain visual grounding behavior in vision-language models. Erel et al. (2026) analyze each attention head as a per-image Markov process over post-softmax attention matrices. Most closely related to our work, Pan et al. (2024) decompose each head's query-key interaction matrix via SVD and qualitatively show that different singular directions attend to distinct semantic relations on hand-picked images.

Why aggregate maps are not enough. Prior work has questioned whether aggregated attention maps faithfully explain transformer computations, both in NLP (Jain & Wallace, 2019; Kobayashi et al., 2020) and vision (Chefer et al., 2021a;b; Abnar & Zuidema, 2020). Additional evidence shows that attention mass can be dominated by sink tokens (Xiao et al., 2024; Darcet et al., 2024), and many

heads are functionally redundant (Voita et al., 2019; Michel et al., 2019), suggesting that head-level aggregation obscures finer computational structure. Existing head-level analyses in ViTs therefore operate on either aggregated attention maps, linearized parameters such as $\mathbf{W}_Q^\top \mathbf{W}_K$ (Pan et al., 2024), or projections into external embedding spaces (Gandelsman et al., 2024; Balasubramanian et al., 2024). In contrast, we directly organize these patterns via a hierarchical tree structure, which enables node-level semantic interpretation of what an attention head computes.

3 Background & Hypothesis

The goal of this study is to uncover and interpret patterns in the query-key space of attention. §3.1 introduces attention and HDBSCAN, and §3.2 presents the central hypothesis of this work.

3.1 Background

Attention weight. We consider a standard Vision Transformer (ViT) architecture (Dosovitskiy et al., 2021) with L layers and H attention heads per layer (Vaswani et al., 2017). Let $\mathbf{X}^{\ell-1} = [\mathbf{x}_0^{\ell-1}, \ldots, \mathbf{x}_{P^2}^{\ell-1}] \in \mathbb{R}^{(1+P^2)\times D}$ denote the input tokens to layer ℓ, where P is the number of image patches per side and $\mathbf{x}_0^{\ell-1}$ is the `[CLS]` token. The output of the h-th attention head at layer ℓ for token i is defined as:

$$\text{Attn}^{\ell,h}(\mathbf{x}_i; \mathbf{X}^{\ell-1}) \coloneqq \sum_{j=0}^{P^2} a_{ij}^{\ell,h}\, \mathbf{x}_j^{\ell-1} \mathbf{W}_V^{\ell,h} \mathbf{W}_O^{\ell,h} \in \mathbb{R}^D, \tag{1}$$

where $\mathbf{W}_V^{\ell,h} \in \mathbb{R}^{D\times D_h}$ and $\mathbf{W}_O^{\ell,h} \in \mathbb{R}^{D_h\times D}$ are the value and output projection matrices, respectively, and $D_h \coloneqq D/H$ is the per-head dimensionality. The attention weight $a_{ij}^{\ell,h}$ between query token i and key token j is computed via a scaled dot-product followed by a softmax:

$$a_{ij}^{\ell,h} \coloneqq \text{Softmax}\left(\frac{\mathbf{q}_i^{\ell,h}\cdot \mathbf{k}_0^{\ell,h}}{\sqrt{D_h}}, \ldots, \frac{\mathbf{q}_i^{\ell,h}\cdot \mathbf{k}_{P^2}^{\ell,h}}{\sqrt{D_h}}\right)_j \in \mathbb{R}, \tag{2}$$

where $\mathbf{q}_i^{\ell,h} \coloneqq \mathbf{x}_i^{\ell-1}\mathbf{W}_Q^{\ell,h} \in \mathbb{R}^{D_h}$ and $\mathbf{k}_j^{\ell,h} \coloneqq \mathbf{x}_j^{\ell-1}\mathbf{W}_K^{\ell,h} \in \mathbb{R}^{D_h}$ denote the *query vector* and *key vector*, respectively.

Clustering. **HDBSCAN** (Campello et al., 2013) is a density-based clustering algorithm that identifies clusters of varying shapes and densities by converting DBSCAN into a hierarchical approach. The primary hyperparameter, `min_cluster_size`, specifies the minimum number of samples required to form a distinct cluster, effectively controlling the resolution of the tree's nodes. To assess the quality of the resulting partitions, we utilize the **silhouette score** (Rousseeuw, 1987), a metric ranging from -1 to 1 that quantifies how well a point fits its assigned cluster relative to neighboring clusters. Finally, points that do not satisfy the density criteria are labeled as **noise**, representing outliers that are excluded from the formal cluster hierarchy to maintain semantic purity.

3.2 Hypothesis

A common approach to interpreting an attention head (ℓ, h) is to visualize its attention map, e.g., by fixing the query to the `[CLS]` token and inspecting $\{a_{0j}^{\ell,h}\}_{j=1}^{P^2}$ (Dosovitskiy et al., 2021; Caron et al., 2021; Chefer et al., 2021a). While this reveals how the head aggregates information across image regions, raw attention weights are often noisy and unreliable as explanations of model behavior. Various methods have been proposed to address this issue (Abnar & Zuidema, 2020; Chefer et al., 2021a;b; Kobayashi et al., 2020), yet they still interpret each head as a single aggregated map, potentially obscuring heterogeneous underlying patterns.

We instead start from a simple property: *identical attention weights can arise from distinct underlying causes.* From Equation (2), the attention score is computed as a single scalar $\mathbf{q}_i^{\ell,h}\cdot\mathbf{k}_j^{\ell,h}$, so many different pairs $(\mathbf{q}_i^{\ell,h}, \mathbf{k}_j^{\ell,h})$ can produce the same value. This degeneracy raises the possibility that a single attention weight $a_{ij}^{\ell,h}$ simultaneously encodes multiple, qualitatively different query-key interaction patterns. A natural and straightforward way to directly analyze these underlying patterns is to concatenate $\mathbf{q}_i^{\ell,h}$ and $\mathbf{k}_j^{\ell,h}$, which we refer to as a *QK feature*. We hypothesize that the QK features

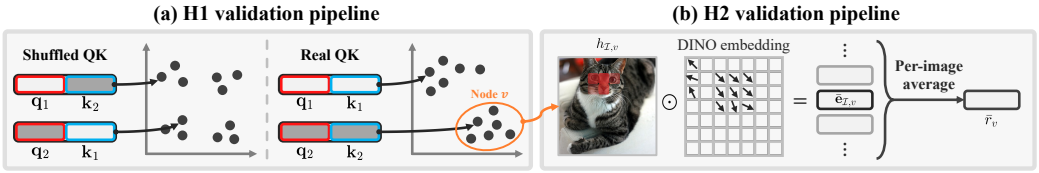


Figure 2: Pipelines for validating two hypotheses. (a) We visualize the distributions of real and shuffled data using a toy example. (b) Using $h_{I,v}$ as weights, we compute the weighted average of DINO embeddings, $\bar{\mathbf{e}}_{I,v}$. A subset of DINO embeddings is visualized as arrows, illustrating that semantically similar patches have similar embeddings.

are composed of multiple distinguishable patterns, each of which carries a consistent semantic role. Concretely, for a fixed head (ℓ, h), we hypothesize that:

- **(H1: structured)** QK feature, the concatenation of query and key vectors, is not a single cloud but is composed of multiple distinguishable patterns.
- **(H2: meaningful)** Each of these patterns carries a consistent semantic role, rather than arising from incidental statistics of the input distribution.

§4 establishes H1 by showing that QK features form multiple distinguishable patterns rather than a single homogeneous distribution. Subsequently, §5 then establishes H2 by organizing these patterns into a hierarchical tree and showing that each node captures a consistent semantic role. Moreover, §G provides causal evidence that these patterns make distinct functional contributions to head behavior.

4 Validating H1: QK Features Form Distinct Patterns

Prior works (Pan et al., 2024; Doshi et al., 2026) have already suggested that query-key patterns can be grouped into qualitatively distinct functional roles. In this section, we go beyond qualitative observations and provide quantitative evidence that such structure indeed exists. We expect that this provides a principled foundation for analyzing attention heads at the level of query-key interactions, and facilitates future methods that exploit this structure.

Setup. We analyze three ViT backbones with different training objectives: supervised ViT-B/16 (Dosovitskiy et al., 2021), CLIP ViT-B/16 (Radford et al., 2021), and DINOv3 ViT-B/16 (Siméoni et al., 2025). For each head (ℓ, h), we forward ImageNet validation images (Russakovsky et al., 2015) and compute $a_{ij}^{\ell,h}$. To restrict the analysis to the most informative interactions, we apply the parameter-free L-method (Salvador & Chan, 2004) to the sorted attention curve and obtain a per-head threshold $\tau^{\ell,h}$ (For details, see §C). We retain only the cells with $a_{ij}^{\ell,h} > \tau^{\ell,h}$ and uniformly sample query-key pairs across the retained attention range. From this pool we form the QK feature $\{\mathbf{z}_n\}_{n=1}^{N}$ (with $N = 10{,}000$), where each $\mathbf{z}_n := [\mathbf{q}_{i_n}^{\ell,h}; \mathbf{k}_{j_n}^{\ell,h}] \in \mathbb{R}^{2D_h}$ is the per-dimension z-scored, ℓ_2-normalized concatenation of one query and one key vector. We compare the empirical distribution of $\{\mathbf{z}_n\}$ against four baseline distributions that progressively corrupt the structure of the real data (see Figure 2a):

- (**Shuffled**) Query and key vectors are individually preserved, but the original $\mathbf{q}$-$\mathbf{k}$ pairing is destroyed, allowing queries and keys from different images to be paired.
- (**Cov-matched**) Each $\mathbf{z}_n$ is generated from a multivariate Gaussian with mean and covariance matched to those of the original $\mathbf{z}_n$ distribution.
- (**Dim-permuted**) The coordinates of each $\mathbf{z}_n$ are randomly permuted independently.
- (**Random**) Each $\mathbf{z}_n$ is sampled independently from an isotropic Gaussian distribution.

Metrics. First, to quantify the structure of $\{\mathbf{z}_n\}$, we follow an established UMAP–HDBSCAN clustering pipeline (Grootendorst, 2022; Angelov, 2020). We reduce the features to a 10-dimensional[1] UMAP embedding, apply HDBSCAN (McInnes et al., 2018; Campello et al., 2013), and report the silhouette score and the fraction of points rejected as noise. Second, we test whether the QK features exhibit multi-modal structure rather than forming a single unimodal cloud using Hartigan's

[1]The UMAP documentation recommends reducing to ~10 dimensions when clustering.

Table 1: Evidence that QK features contain multiple distinct patterns. Each value is the average across 144 heads. Best values in each cell are shown in bold.

Backbone	Metric	Real	Shuffled	Cov-matched	Dim-permuted	Random
ViT	Silhouette (↑)	**0.804**	0.796	0.550	0.156	0.000
	HDBSCAN noise (↓)	**0.026**	0.028	0.547	0.644	1.000
	Splittable frac. (↑)	**0.639**	0.500	0.014	0.000	0.000
CLIP	Silhouette (↑)	**0.806**	0.799	0.597	0.118	0.000
	HDBSCAN noise (↓)	**0.035**	0.046	0.499	0.818	1.000
	Splittable frac. (↑)	**0.681**	0.632	0.014	0.000	0.000
DINOv3	Silhouette (↑)	**0.825**	0.811	0.560	0.182	0.000
	HDBSCAN noise (↓)	**0.043**	0.063	0.539	0.602	1.000
	Splittable frac. (↑)	**0.861**	0.729	0.007	0.000	0.000

dip test (Hartigan & Hartigan, 1985), a standard non-parametric test of unimodality (Freeman & Dale, 2013; Maurus & Plant, 2016). Concretely, we apply Dip-means (Kalogeratos & Likas, 2012), which recursively splits the data only when the test finds statistically significant evidence of multiple modes. We report the Splittable fraction, defined as the proportion of heads where more than one mode is detected. If QK features possess underlying structure, the real data should exhibit high silhouette scores and Splittable fractions and a low noise ratio, with these signals degrading under corruption. Otherwise, no significant differences should be observed between real and corrupted data.

Results. Table 1 summarizes the results. Consistent with our expectation, real QK features achieve the strongest scores across all metrics, with values degrading progressively as the data undergoes stronger corruption. Notably, the *shuffled* baseline still achieves relatively high scores because the query and key representations themselves remain preserved. However, real QK features consistently perform better, indicating that the query-key pairing contributes meaningful additional structure beyond the individual representations alone. Overall, all three diagnostics provide strong evidence for Hypothesis H1: *real query-key paired features (i.e., QK features) cannot be explained by simple null distributions, and are instead composed of multiple distinguishable patterns.* Full results with standard deviations and additional metrics are reported in §D.

5 Validating H2: QK Trees Reveal Consistent Semantic Roles

In §4, we demonstrated that the QK feature distribution is not merely a single cloud but instead possesses a specific underlying structure. Building upon this observation, we now present evidence supporting Hypothesis H2. First, §5.1 describes our method to organize QK features into a tree structure via recursive clustering. Then, §5.2 provides a quantitative analysis demonstrating that these nodes exhibit consistent semantic coherence.

5.1 Building a Tree from QK Features

We utilize the QK features $\{\mathbf{z}_n\}_{n=1}^{N}$ from §4, increasing the sample size to $N = 50{,}000$ for cluster resolution. We again project $\{\mathbf{z}_n\}$ onto a 10-dimensional UMAP embedding and cluster with HDBSCAN. To avoid manually fixing the `min_cluster_size` hyperparameter, we evaluate six candidate values and select the partition with the highest silhouette score among those producing at least two clusters. If the peak silhouette score remains below the "reasonable structure" threshold of 0.5 (Rousseeuw, 1987), the node is designated as a leaf. Otherwise, the same procedure is applied recursively (Figure 1b). This recursive procedure yields a hierarchical **QK tree** $\mathcal{T}^{\ell,h}$ for each attention head (ℓ, h). We set $d_{\max} = 3$, since most depth-3 nodes contain insufficient QK features for further splitting (§C). Note that each node in a QK tree represents a distinct subset of QK features and therefore serves as the basic unit of analysis in the remainder of this work. The formal algorithm is provided in §C.

To enable the assignment of an unseen feature $\mathbf{z}_{\text{test}}$ to a corresponding leaf node, we train a lightweight random-forest classifier $\mathcal{C}^v$ at each internal node $v \in \mathcal{T}^{\ell,h}$. During inference, the model performs a top-down traversal from the root, evaluating a single classifier at each hierarchy level to determine the

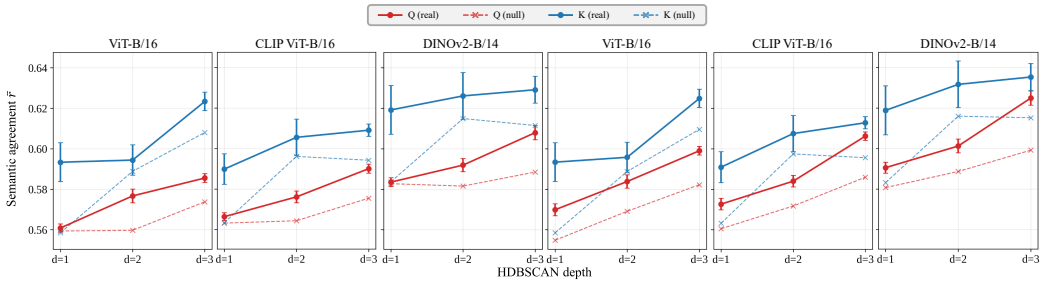


Figure 3: Semantic agreements as a function of QK tree depth across three backbones. Within each panel, $\bar{r}^{\mathrm{Q}}$ (●) and $\bar{r}^{\mathrm{K}}$ (●) are shown for each non-root node v in two forms: real ($\bar{r}^{\mathrm{Q}}_{v}$, solid) and parent-matched null ($\bar{r}^{\mathrm{Q}}_{u}$ where $u = \mathrm{parent}(v)$, dashed), both evaluated on v's prototype images.

final cluster membership. Thus, given a QK feature $\mathbf{z}_n = [\mathbf{q}_i; \mathbf{k}_j]$, the chain of classifiers $\{\mathcal{C}^v\}$ along the root-to-v path determines whether $\mathbf{z}_n$ is routed to node v, yielding the membership indicator:

$$\mathrm{mem}_v(\mathbf{q}_i; \mathbf{k}_j) \;\coloneqq\; \mathbb{1}\big[\text{ the path classifiers route } [\mathbf{q}_i; \mathbf{k}_j] \text{ to node } v\,\big] \;\in\; \{0, 1\}. \tag{3}$$

5.2 Quantitative Analysis of Cluster Consistency

To establish Hypothesis H2, this section shows that each node of the QK tree encodes a consistent semantic role. Concretely, our goal is to verify that, for QK features assigned to a node v, the corresponding queries and keys are located on semantically consistent regions, respectively.

Frequency maps. For a given image I and all image patch indices $i, j \in \{0, \ldots, P^2\}$, we define the following two frequency maps:

$$\tilde{h}^{\mathrm{Q}}_{I,v}(i) \coloneqq \frac{\sum_j 1 \cdot \mathrm{mem}_v(\mathbf{q}_i, \mathbf{k}_j)}{\sum_j 1} \in [0, 1], \qquad h^{\mathrm{Q}}_{I,v}(i) \coloneqq \frac{\sum_j \alpha_{i,j}\, \mathrm{mem}_v(\mathbf{q}_i, \mathbf{k}_j)}{\sum_j \alpha_{i,j}} \in [0, 1]. \tag{4}$$

Intuitively, $\tilde{h}^{\mathrm{Q}}_{I,v}(i)$ represents the fraction of key positions j for which the pair $(\mathbf{q}_i, \mathbf{k}_j)$ is assigned to node v, while $h^{\mathrm{Q}}_{I,v}(i)$ is its attention-weighted counterpart, emphasizing pairs with higher attention. For each image I, we compute $\max_i \tilde{h}^{\mathrm{Q}}_{I,v}(i)$ (or $\max_i h^{\mathrm{Q}}_{I,v}(i)$) and define the images with the largest values as the *Q-prototype images* of node v. We analogously define the *K-prototype images* using the K-side frequency maps. Note that although the QK tree is constructed from QK features whose attention weights exceed the L-method threshold, we extend mem_v to all attention pairs during analysis. A justification for this extension is provided in §E.

Preliminary: DINO embeddings. Our goal is to test whether each node v encodes a consistent semantic role. We leverage a key property of self-supervised DINO features (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025; Amir et al., 2021): patches with similar visual content are mapped to nearby points on the unit hypersphere. As a result, cosine similarity between patch embeddings serves as a reliable proxy for semantic similarity (Hamilton et al., 2022). Following Pach et al. (2026), we use this property to evaluate whether the activations $\tilde{h}^{\mathrm{Q}}_{I,v}(i)$, $h^{\mathrm{Q}}_{I,v}(i)$, $\tilde{h}^{\mathrm{K}}_{I,v}(i)$, and $h^{\mathrm{K}}_{I,v}(i)$ consistently focus on semantically similar regions, as described in the next paragraph.

Method. We describe the procedure using $h^{\mathrm{Q}}_{I,v}(i)$; the same method applies to the other cases. For each of the top k prototype images (default $k = 10$), we first normalize the DINOv3 (Siméoni et al., 2025) patch embeddings onto the unit hypersphere, yielding $\mathbf{e}_I \in \mathbb{R}^{P \times P \times D}$. Using $h^{\mathrm{Q}}_{I,v} \in \mathbb{R}^{P \times P}$ as weights, we then compute a weighted average embedding vector $\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v} \in \mathbb{R}^{D}$ for each image (Figure 2b). That is, the more strongly $h^{\mathrm{Q}}_{I,v}$ concentrates on semantically similar regions, the larger $\|\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}\|$ becomes. We aggregate these vectors across the set $\mathcal{I}$ of Q-prototype images and define $\bar{r}^{\mathrm{Q}}_{v} \coloneqq \frac{1}{|\mathcal{I}|} \big\|\sum_{I \in \mathcal{I}} \bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}\big\|$, where $|\mathcal{I}| = k$. Thus, $\bar{r}^{\mathrm{Q}}_{v} \in [0, 1]$ is large when the activated regions are semantically concentrated within images and consistent across images. We compute $\bar{r}^{\mathrm{Q}}_{v}$ for ViT-B/16, CLIP ViT-B/16, and DINOv2-B/14[2], and report the average over all nodes v along root-to-leaf paths across all $\mathcal{T}^{\ell,h}$. Full mathematical details are provided in §C.5.

[2]We report DINOv2 rather than DINOv3 here, since evaluating DINOv3 frequency maps against its own embeddings would not provide an independent test.

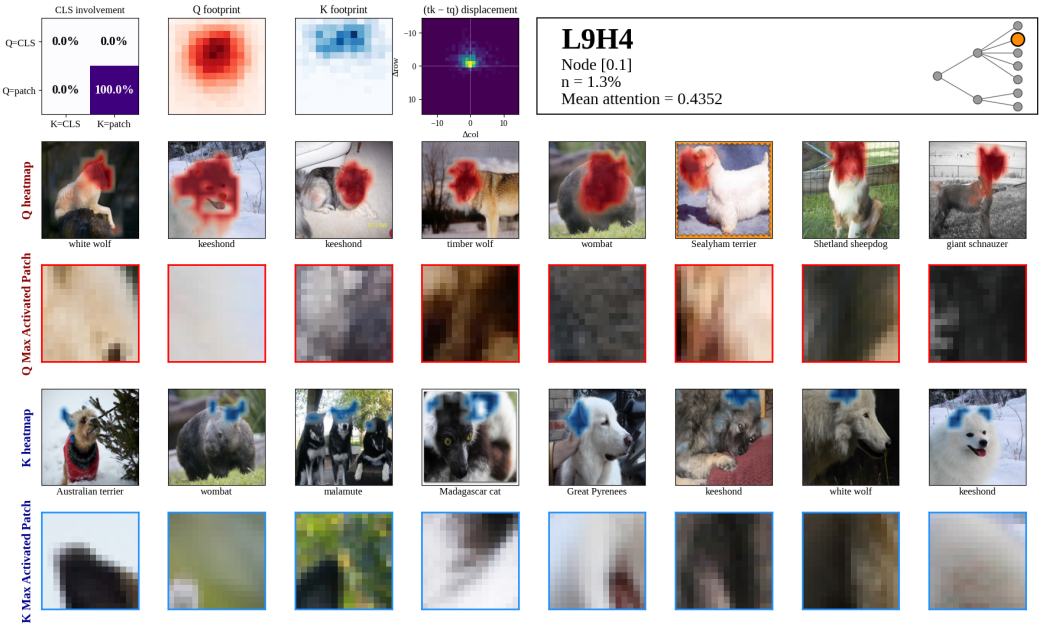


Figure 4: Example role card for a node $[0.1]$ in ViT ($\ell = 9, h = 4$). A QK feature is assigned to this node when the query (•) is located on the animal's head and the key (•) is located on its ear.

Parent-matched null. A high value of $\bar{r}_v^{\mathrm{Q}}$ may arise not only from semantic consistency, but also from a favorable selection of prototype images. For example, if the Q-prototype images of v are mostly close-up views of the same object category, even a spatially unselective map may yield a high $\bar{r}_v^{\mathrm{Q}}$, since many patch embeddings within those images are already similar. To isolate the contribution of node v from this image-selection effect, we introduce a *parent-matched null*. Let u denote the parent of v. Since the QK features assigned to v form a subset of those assigned to u, the parent map covers the region associated with v together with those of its sibling nodes. We compute $\bar{r}_u^{\mathrm{Q}}$ using the same procedure as above, but on the Q-prototype images of v. If node v captures finer semantic structure than its parent, we expect $\bar{r}_v^{\mathrm{Q}} > \bar{r}_u^{\mathrm{Q}}$.

Results. The results are presented in Figure 3. The left three panels show $\bar{r}$ computed using $\tilde{h}_{I,v}^{\mathrm{Q}}(i)$ and $\tilde{h}_{I,v}^{\mathrm{K}}(i)$, while the right three panels show $\bar{r}$ computed using $h_{I,v}^{\mathrm{Q}}(i)$ and $h_{I,v}^{\mathrm{K}}(i)$. Across all conditions, both $\bar{r}^{\mathrm{Q}}$ and $\bar{r}^{\mathrm{K}}$ increase monotonically with depth, and consistently exceed their parent-matched null counterparts at every level. This indicates that each node captures semantically coherent regions, and that deeper nodes provide progressively finer semantic specialization beyond their parents. A layer-wise breakdown and k ablation study are provided in §F. Taken together, these results support Hypothesis H2: *nodes in the QK tree exhibit strong semantic consistency, with clearer structure emerging at greater depths.*

6 Interpreting a Node with a Role Card

§4 and §5 establish that each node in the QK tree encodes a consistent semantic role. We now turn to interpreting what these roles are. First, §6.1 introduces the *role card*, a diagnostic visualization tool for interpreting individual nodes in the QK tree. Then, §6.2 uses it to identify *position-indicator nodes*, showing that spatial selectivity is localized within specific nodes, not across an entire head.

6.1 Role Card: A New Interpretation Tool for Attention Heads

A role card is a visual diagnostic tool that characterizes the functional role of a node in the QK tree. It consists of multiple panels, each capturing a different aspect of the QK feature cluster associated with a node v. As an example, Figure 4 presents the role card of a node from the ViT head ($\ell = 9, h = 4$).

This node corresponds to the second child of the first child of the root; using zero-based indexing, we denote it as node $[0.1]$.

Panels of a role card. The two maps $h^{\mathrm{Q}}_{I,v}$ and $h^{\mathrm{K}}_{I,v}$ are rendered as the second and fourth rows (`Q/K heatmap`) of the role card on the top 8 Q-prototype images and the top 8 K-prototype images of v, respectively. The third and fifth rows (`Q/K Max Activated Patch`) visualize the patch at a position yielding $\max_i h^{\mathrm{Q}}_{I,v}(i)$ and $\max_i h^{\mathrm{K}}_{I,v}(i)$. To summarize the typical behavior of node v, four additional panels are shown in the top row. The `CLS involvement` panel reports the distribution of query-key pairs assigned to v over the four combinations of token types (image patch vs. non-image token). For instance, the node in Figure 4 exclusively captures cases where both the query and key correspond to image patches, and is therefore labeled as (Q = patch, K = patch) with 100%. The `Q footprint` and `K footprint` panels display the per-image means of $h^{\mathrm{Q}}_{I,v}$ and $h^{\mathrm{K}}_{I,v}$, averaged over the 50 candidate images of v. The `(tk-tq) displacement` panel shows the relative position of the key patch with respect to the query patch, revealing whether the node tends to capture local, fixed-offset, or global spatial relationships. Details on the definition of each panel are provided in §H.1.

Example: Interpreting ViT $(\ell = 9, h = 4)$ **head.** Figure 4 shows a role card visualizing node $[0.1]$ from $\mathcal{T}^{9,4}$ in ViT. This node captures patterns where the query attends to an animal's head while the key corresponds to its ear (`animal head` $\rightarrow$ `animal ear`). The spatial distribution of the `K footprint` lies above that of the `Q footprint`, and the `(tk-tq) displacement` panel is skewed upward from the center, further supporting this interpretation at the aggregate level. While this example illustrates a single node, a broader pattern such as `human torso` $\rightarrow$ `human arm` emerges when examining other nodes in $\mathcal{T}^{9,4}$ (§H.2). In addition, nodes at greater depths consistently exhibit more specific and semantically coherent behaviors, indicating that interpretability improves as we move down the tree, supporting the result of Figure 3. Thus, our hierarchical refinement facilitates a more structured analysis of attention behavior, with coarse patterns near the root and more specialized roles at deeper levels. Additional role card examples for other attention heads are provided in §H.2.

6.2 Finding: Position Indicators in Attention Heads

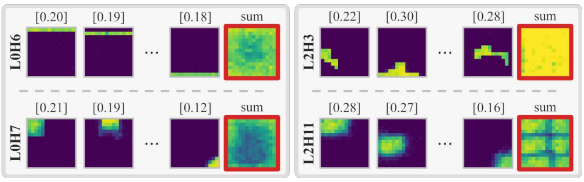


Figure 5: Example position indicator nodes in ViT.

Voita et al. (2024) introduce an automatic procedure for identifying language-model features that activate at fixed token positions. Kim et al. (2025) extend this approach to Vision Transformers, recovering positional indicators in the residual stream. Following them, we determine whether the Q heatmaps of a node are positional by binarizing each heatmap at half of its maximum and computing the mutual information between activation and patch position. Figure 5 shows the binarized Q heatmaps of positional indicators found in ViT-B/16, averaged over images. Three findings emerge. ① Identified positional nodes correspond to distinct sub-regions of the patch grid, and the union of positions covered by nodes within a single head spans the entire image, extending the previous observations of Kim et al. ② The shapes of position indicators vary across heads, indicating that each head partitions the grid into qualitatively distinct sub-regions. ③ These shapes become progressively more complex with tree depth, suggesting that deeper nodes capture richer spatial selectivity. Additional results for K heatmaps, other backbones and heads, along with further discussion, are provided in §I.

7 Node-Level Selection Improves Object Localization

Existing attention-based methods typically rely on head-level signals, such as `[CLS]`-to-patch attention or aggregates across heads. However, §5 shows that each head contains multiple semantically distinct QK patterns, suggesting that only a subset of its nodes may carry object-relevant information. In this section, we show that this finer-grained view is *actionable* by comparing object localization at three levels of granularity: *layer*, *head*, and *node*, from coarser to finer.

Table 2: IoU results on ECSSD / DUT-OMRON / DUTS-TE (higher is better). All four rows read the same attention of the same frozen backbone and differ only in what is selected: nothing, one layer, one head, one node. The three selected rows use the same 500 calibration images and the same layer range. Bold and underline mark the best and second-best per cell, respectively.

Method	ViT-B/16	CLIP B/16	DINOv2+reg B/14	DINOv3 B/16
Attention map	.152 / .130 / .096	.078 / .072 / .079	.383 / .253 / .339	.380 / .254 / .346
Best layer	.438 / <u>.290</u> / **.354**	<u>.375</u> / <u>.255</u> / <u>.307</u>	.500 / <u>.272</u> / .374	<u>.485</u> / <u>.271</u> / .374
Best head	<u>.451</u> / .283 / .340	.373 / .187 / .295	<u>.535</u> / .248 / <u>.390</u>	.472 / .248 / <u>.399</u>
Ours (one node)	**.469 / .336** / <u>.349</u>	**.569 / .400 / .435**	**.647 / .448 / .481**	**.682 / .449 / .521**

Node selection method. For each of four frozen backbones, supervised ViT-B/16 (Dosovitskiy et al., 2021), CLIP ViT-B/16 (Radford et al., 2021), DINOv2+reg ViT-B/14 (Oquab et al., 2024), and DINOv3 ViT-B/16 (Siméoni et al., 2025), we consider QK tree nodes from layers 6–11. For each node $v \in \mathcal{T}^{\ell,h}$, we use its Q-frequency map[3] over image tokens, $h^{\mathrm{Q}}_{I,v} \in [0,1]^{P\times P}$ (§5.2), as the localization map. We threshold the map at 0.5, a standard choice for normalized mask probabilities (Cheng et al., 2021; 2022), and compute its IoU with the corresponding saliency mask on 500 DUTS-TR (Wang et al., 2017) images. We select the node with the highest mean IoU and evaluate it unchanged on ECSSD (Yan et al., 2013), DUT-OMRON (Yang et al., 2013), and DUTS-TE (Wang et al., 2017).

Baseline methods. For each backbone, we search for the best *head* and *layer* over the same layers 6–11, using the same DUTS-TR selection set and evaluating on the same test datasets. The head baseline treats each `[CLS]`-to-patch attention map from a layer-head pair as one candidate, while the layer baseline averages these maps over the twelve heads of a layer. To give these baselines additional flexibility, we allow each candidate to select its binarization rule, either by keeping the patches whose attention mass sums to a fraction $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ of the total (Caron et al., 2021), or by thresholding at the map mean. In contrast, the node-level method uses a fixed threshold of 0.5. We retain the candidate with the highest mean IoU and apply it unchanged at test time. For reference, we also report the last-layer `[CLS]`-to-patch attention map averaged over heads, with no selection. Full details are provided in §J.1.

Results. Table 2 reports the comparison. Node-level selection performs best in nearly all settings, with the only exception being supervised ViT on DUTS-TE. This is consistent with the fact that a head or layer aggregates multiple QK patterns, whereas a node isolates a single pattern. When object-relevant information is already concentrated within a head or layer, such aggregation can itself provide a strong localization signal. Note that a single node is not the strongest setting of our method. We use one node here to isolate the effect of granularity, while combining multiple nodes further improves localization across backbones (§K.2). Overall, these results support the view that attention heads are not functionally homogeneous, and that their internal QK structure can expose object-relevant signals obscured by coarser aggregation.

8 Conclusion

We revisited attention in Vision Transformers by analyzing query-key interactions rather than aggregated attention weights. By defining QK features and organizing them via recursive clustering, we introduced the QK tree, a hierarchical decomposition of attention heads into functionally distinct nodes. We showed that QK features exhibit structured, non-unimodal distributions, and that each node encodes a consistent semantic role, with deeper nodes providing finer specialization. To facilitate interpretation, we proposed the role card for visualizing node behavior, and identified position-indicator nodes that reveal spatial selectivity localized within specific nodes. Finally, we demonstrated that this decomposition is actionable by deriving a simple and effective object localization method. We hope this work motivates further study of attention at finer levels, beyond aggregated attention weights. We discuss the limitations of our work in §A.

[3]By default, we use $h^{\mathrm{Q}}_{I,v}$ in place of $\tilde{h}^{\mathrm{Q}}_{I,v}$ due to its stronger $\bar{r}$, and in place of $h^{\mathrm{K}}_{I,v}$ due to its lower variance. We provide an ablation of these choices in §K.4.

AI USE STATEMENT

In this work, we used generative AI tools for partial translation and language polishing of the manuscript, organizing existing code, and checking consistency between the manuscript and the implementation. We did not use generative AI tools to generate research ideas, design or specify the experimental methodology, write substantive experimental code, or run experiments. Other required disclosure tasks are not applicable to this work. We reviewed all AI-assisted work. All research ideas, experimental designs and refinements, implementation decisions, experiments, and scientific conclusions were developed and carried out by the authors. AI-assisted manuscript edits were reviewed by the authors, and AI-assisted code organization and manuscript–code consistency checks were verified against the authors' implementation and experimental procedures. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the full implementation used in this work as supplementary material. Detailed descriptions of our method, including the recursive clustering algorithm, implementation details, experimental protocols, and evaluation settings, are provided in §C. All metrics used in the main text are described in §D.1, and the procedure for identifying positional indicator nodes in §6.2 is detailed in §I.1. Together, these materials provide the information needed to reproduce the main experiments and analyses in this work.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

Dimo Angelov. Top2Vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. Decomposing and interpreting image representations via text in ViTs beyond CLIP. *Advances in Neural Information Processing Systems*, 37:81046–81076, 2024.

Amit Banerjee and Rajesh N. Davé. Validating clusters using the Hopkins statistic. In *IEEE International Conference on Fuzzy Systems*, pp. 149–153, 2004.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von Mises–Fisher distributions. *Journal of Machine Learning Research*, 6: 1345–1382, 2005.

Adam Bielski and Paolo Favaro. MOVE: Unsupervised movable object segmentation and detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

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

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 (PAKDD)*, pp. 160–172, 2013.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

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*, pp. 9650–9660, 2021.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 782–791, June 2021a.

Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *International Conference on Computer Vision (ICCV)*, 2021b.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pp. 276–286, 2019.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.

Fenil R Doshi, Thomas Fel, Talia Konkle, and George Alvarez. Bi-orthogonal factor decomposition for vision transformers. *arXiv preprint arXiv:2601.05328*, 2026.

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. *ICLR*, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Yotam Erel, Olaf Dünkel, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Amit Bermano. Attention (as discrete-time markov) chains. *Advances in Neural Information Processing Systems*, 38:54658–54690, 2026.

Jonathan B Freeman and Rick Dale. Assessing bimodality to detect the presence of a dual cognitive process. *Behavior research methods*, 45(1):83–97, 2013.

Yossi Gandelsman, Alexei Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *International conference on learning representations*, volume 2024, pp. 18395–18416, 2024.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, 2023.

Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*, 2022.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. Pmlr, 2018.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

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

Sonia Joseph, Praneet Suresh, Lorenz Hufe, Edward Stevinson, Robert Graham, Yash Vadi, Danilo Bzdok, Sebastian Lapuschkin, Lee Sharkey, and Blake Aaron Richards. Prisma: An open source toolkit for mechanistic interpretability in vision and video. *arXiv preprint arXiv:2504.19475*, 2025b.

Argyris Kalogeratos and Aristidis Likas. Dip-means: An incremental clustering method for estimating the number of clusters. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jinyeong Kim, Junhyeok Kim, Yumin Shim, Joohyeok Kim, Sunyoung Jung, and Seong Jae Hwang. Interpreting vision transformers via residual replacement model. *arXiv preprint arXiv:2509.17401*, 2025.

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. Interpreting attention layer outputs with sparse autoencoders. *arXiv preprint arXiv:2406.17759*, 2024.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Ruosen Li, Teerth Patel, and Xinya Du. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*, 2023.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, 2024.

Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Wiley, 2000.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, volume 2025, pp. 23888–23923, 2025.

Samuel Maurus and Claudia Plant. Skinny-dip: clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1055–1064, 2016.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Matthew Lyle Olson, Musashi Hinck, Neale Ratzlaff, Changbai Li, Phillip Howard, Vasudev Lal, and Shao-Yen Tseng. Probing the representational power of sparse autoencoders in vision models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6226–6236, 2025.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. *Advances in Neural Information Processing Systems*, 38:95706–95742, 2026.

Xu Pan, Aaron Philip, Ziqian Xie, and Odelia Schwartz. Dissecting query-key interaction in vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

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.

Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *16th IEEE international conference on tools with artificial intelligence*, pp. 576–584. IEEE, 2004.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *Proceedings of the British Machine Vision Conference (BMVC)*, November 2021.

Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonin Vobecký, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3, 2025.

Michael Steinbach. A comparison of document clustering techniques. In *KDD workshop on text mining*, 2000.

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

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *arXiv preprint arXiv:2605.29358*, 2026.

Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3554–3563. IEEE, 2024.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1288–1301, 2024.

Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7486–7496. IEEE, 2023.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, 45(12):15790–15801, 2023b.

Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. Gradient based feature attribution in explainable ai: A technical review. *arXiv preprint arXiv:2403.10415*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18113–18123. IEEE, 2022a.

Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4300–4309. IEEE, 2022b.

Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3166–3173, 2013.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with hierarchical sparse autoencoders. *arXiv preprint arXiv:2502.20578*, 2025.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

Hasib Zunair and A. Ben Hamza. PEEKABOO: Hiding parts of an image for unsupervised object localization. In *British Machine Vision Conference (BMVC)*, 2024.

Appendix: Table of Contents

Attention Weight as a Hierarchical Tree in ViT

A Limitations **18**

B Additional Related Work **18**

C Method Details **19**

C.1 L-method 19

C.2 Balanced-bin Allocation 20

C.3 Recursive Clustering 21

C.4 Per-node Classifiers 22

C.5 Formal Definitions for Cluster Consistency 22

D Full Evidence of H1 **24**

D.1 Metrics and the Dip-means Procedure 24

D.2 Full Results 25

E Justification of Pipeline Design Choices **26**

E.1 L-method Ablation: Full-range Sampling Substantially Weakens Clustering 27

E.2 Generalization from Highonly Training to All-Pair Evaluation 27

E.3 Human Preference Study on Role-Card Interpretability 28

F Supplementary Cluster-Consistency Analyses **30**

F.1 Layer-Grouped Trends 31

F.2 Robustness to Prototype Count k 31

G Causal Relevance of Nodes **32**

H Role Cards **34**

H.1 Details on Each Panel 34

H.2 Role Card Examples 34

I Positional Structure in Early-Layer QK Trees **84**

I.1 Positional-leaf Criterion and Coverage Metrics 84

I.2 Experimental Setup for Positional Analysis 84

I.3 Positional Coverage across Backbones 85

I.4 Per-head Visualizations of Positional Leaves 85

J Object Localization Details **102**

J.1 Detailed Experimental Setup 102
J.2 Qualitative Results 102

K Top k Sweep and Inference-Time Trade-off for Object Localization **102**

K.1 Greedy Multi-node Selection 102
K.2 Full k-sweep 103
K.3 Per-k Inference-time Benchmark 104
K.4 Q-side vs. K-side Aggregation 106

A LIMITATIONS

Due to space constraints, we leave several limitations of our study to this section rather than the main paper. We discuss them from three perspectives: ① the supervision required for node selection, ② the applicability of discrete clustering across attention heads, ③ and the dependence of the semantic-consistency metric on the choice of probe representation.

Node selection requires labeled masks. The localization method of §7 is training-free in the sense that no backbone parameters are updated, but it is not label-free. The node is selected by scoring candidates against the ground-truth saliency masks of 500 DUTS-TR images. This calibration is performed once per backbone, but it still requires supervision that fully unsupervised object-discovery methods do not. Whether a comparably effective node can be identified using an unsupervised criterion, such as the semantic-consistency statistic of §5.2, remains an open question.

Not every head decomposes into clusters. The QK tree is most appropriate when the QK features of a head form separable density modes. This condition holds for many heads, but not all. Depending on the backbone, 14–36% of heads are not splittable under the Dip-means criterion (Table 1), and in DINOv3 the early-layer positional structure does not resolve into clusters even after relaxing the splitting hyperparameters (§I). When the underlying variation is closer to continuous than multimodal, a hard partition may be an inappropriate representation. In such cases, the tree either remains unsplit or introduces boundaries that need not correspond to distinct functional roles.

The cluster-consistency metric depends on the probe representation. The semantic-consistency statistic $\bar{r}_v^{\mathrm{Q}}$ introduced in §5 is measured in the embedding space of an external probe model. We use frozen DINOv3 ViT-B/16 embeddings for the ViT, CLIP, and DINOv2 backbones, and frozen DINOv2 ViT-B/14 embeddings for the DINOv3 backbone, so that the probe is always distinct from the dissected backbone (§C.5). The absolute value of $\bar{r}_v^{\mathrm{Q}}$ therefore depends on the choice of probe representation. We mitigate this dependence using the parent-matched null in §C.5, which compares a node with its parent under the same probe rather than comparing absolute values across different probes. Nevertheless, developing a probe-independent measure of semantic consistency remains an open direction.

B ADDITIONAL RELATED WORK

This section expands §2 along three directions, each corresponding to a question raised in the main text: ① whether an attention head should be treated as the smallest meaningful unit of analysis, a premise underlying our decomposition (§4), ② how the QK tree relates to sparse-dictionary methods that address polysemantic representations at a different site in the network (§5.1), and ③ what signals are consumed by training-free object-discovery methods, which motivates the baselines in §7.

Sub-head units of analysis in language models. A long line of work in NLP has examined functional specialization below or across the level of individual attention heads. Clark et al. (2019) report that BERT heads align with specific syntactic and coreference relations, while Voita et al. (2019) and Michel et al. (2019) show that a substantial fraction of heads can be pruned with little or no performance degradation. Mechanistic-interpretability work studies computation in terms of circuits spanning multiple components. Elhage et al. (2021) develop a mathematical framework for attention-only transformers that distinguishes several computational motifs. Olsson et al. (2022) study induction heads as a mechanism associated with in-context learning. Wang et al. (2022) trace an end-to-end circuit for indirect object identification in GPT-2 Small, and Conmy et al. (2023) automate circuit discovery through causal interventions. A parallel line of work analyzes individual feed-forward neurons. Voita et al. (2024) identify "dead," positional, and n-gram-detector neurons in OPT, and Kim et al. (2025) adapt positional-feature analysis to vision transformers. Our work is related in spirit, but focuses specifically on decomposing the query-key interactions within a single attention head.

Superposition and sparse-dictionary decompositions. A related line of work studies polysemantic representations through the lens of superposition. Elhage et al. (2022) show that networks can

represent more features than they have available dimensions by storing those features in superposition. A prominent approach to separating such features is to learn a sparse, over-complete dictionary. Bricken et al. (2023) train sparse autoencoders (SAEs) on MLP activations in a one-layer transformer to recover interpretable features. Huben et al. (2024) extend this approach to the residual stream of Pythia, and Templeton et al. (2026) scale it to Claude 3 Sonnet. Subsequent work studies alternative training objectives and architectures (Gao et al., 2025; Rajamanoharan et al., 2024; Bussmann et al., 2025) and releases SAEs covering many model components (Lieberum et al., 2024; Marks et al., 2025). The same general approach has also been applied in vision, where SAEs trained on frozen ViT and CLIP representations recover interpretable and experimentally testable visual features (Joseph et al., 2025b; Stevens et al., 2025; Olson et al., 2025; Zaigrajew et al., 2025; Joseph et al., 2025a). Closest to our setting, Kissane et al. (2024) apply SAEs to attention-layer outputs and identify feature families associated with long-range, short-range, and induction-like behaviors. Unlike sparse-dictionary methods, the QK tree directly partitions the joint $(\mathbf{q}, \mathbf{k})$ distribution within an attention head.

Training-free object discovery. A broad family of training-free or minimally supervised object-discovery methods builds on frozen self-supervised vision transformers. Amir et al. (2021) demonstrated that DINO-ViT patch embeddings serve as strong dense visual descriptors. LOST (Siméoni et al., 2021) seeds and expands regions using patch-similarity structure. TokenCut (Wang et al., 2023b) and Deep Spectral Methods (Melas-Kyriazi et al., 2022) partition patch-similarity graphs using spectral methods. FOUND (Siméoni et al., 2023) exploits background cues derived from attention. MOVE (Bielski & Favaro, 2022) learns a movability prior over frozen representations, PEEKABOO (Zunair & Ben Hamza, 2024) localizes objects by selectively hiding image regions, and CutLER (Wang et al., 2023a) converts unsupervised proposals into training data for a detector. These methods typically operate on attention-derived maps or patch-feature similarity structures, rather than on individual query-key interactions within a head.

C Method Details

This section provides the implementation details omitted from §4 and §5. Unless otherwise specified, all numerical values correspond to the default settings used for every figure and table in the main paper and match the released implementation.

C.1 L-method

Overview of the L-method. The L-method (Salvador & Chan, 2004) is a parameter-free knee-detection algorithm for monotone curves. Given a sorted non-decreasing sequence $y_1 \le y_2 \le \cdots \le y_n$, indexed by normalized coordinates $x_i = i/n$, the method identifies the index $i^\star$ that best separates the curve into a slowly varying left segment and a rapidly increasing right tail. For each candidate split point i, two linear regressions are fitted independently: one to the first $i+1$ points and one to the remaining $n-i$ points. The optimal split minimizes the length-weighted reconstruction error:

$$J(i) \;=\; \frac{i+1}{n}\,\mathrm{MSE}_{\mathrm{left}}(i) + \frac{n-i}{n}\,\mathrm{MSE}_{\mathrm{right}}(i). \tag{5}$$

The selected index, $i^\star = \arg\min_i J(i)$ corresponds to the elbow of the curve, and the associated value $y_{i^\star}$ is used as a data-driven threshold separating the bulk of the distribution from its long tail. The L-method requires no manually chosen hyperparameters and is invariant to monotone rescaling of the curve. This makes it particularly suitable for attention distributions, whose magnitudes differ substantially across heads.

Motivation for thresholding. Without thresholding, the row-wise softmax attention matrix contains a large number of near-uniform low-magnitude entries. Each query token assigns a small nonzero attention weight to nearly every key token, producing a long left tail dominated by weak interactions. Including these entries in the joint feature pool of §4 and in the clustering procedure of §5.1 substantially degrades the structure of the resulting feature space (we refer the reader to §E). Specifically, the silhouette score decreases, the HDBSCAN noise fraction increases, and the discovered clusters begin to reflect statistics of the softmax tail rather than the high-attention interaction patterns of interest. The L-method threshold therefore provides a simple and principled mechanism for removing this tail while retaining the dominant attention modes.

Implementation details. For each head (ℓ, h), we process 2,000 ImageNet (Russakovsky et al., 2015) validation images through the backbone model, yielding $2{,}000 \times T \times T$ row-wise softmax attention values, where $T = 1 + P^2$ denotes the total number of tokens (one CLS token and P^2 patch tokens). To estimate the threshold efficiently, we uniformly subsample 200,000 attention entries from this pool, sort them in ascending order, and resample the resulting curve at $n = 1{,}000$ equally spaced quantile locations. We then evaluate $J(i)$ for all candidate split indices satisfying $2 \le i \le n - 3$, ensuring that both fitted segments contain at least two points. The final threshold for head (ℓ, h) is defined as $\tau^{\ell,h} := y_{i^\star}$, where $y_{i^\star}$ is the curve value at the optimal split point. Empirically, nearly all heads exhibit a clear elbow structure under the L-method, with the selected threshold consistently separating a dense low-attention regime from a sparse high-attention tail (Figures 6-8).

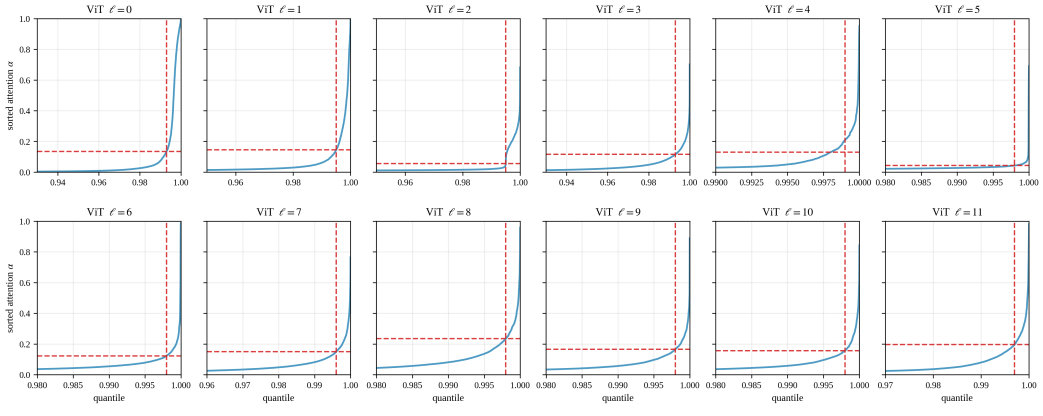


Figure 6: L-method thresholding results for all layers of ViT.

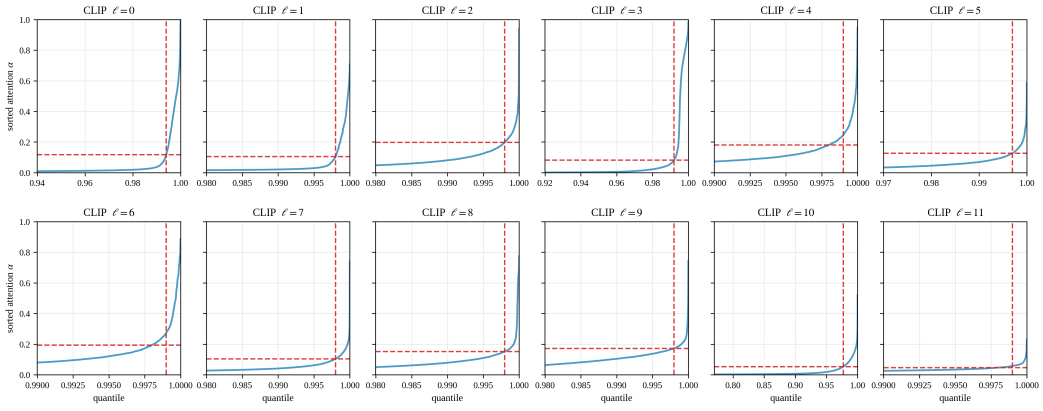


Figure 7: L-method thresholding results for all layers of CLIP.

C.2 Balanced-bin Allocation

Let $\mathcal{A} = \{a_m : a_m \ge \tau^{\ell,h}\}$ denote the set of above-threshold attention entries collected from the 2,000 calibration images. We partition $\mathcal{A}$ into $B = 10$ percentile bins with boundaries

$$\tau^{\ell,h} = b_0 < b_1 < \cdots < b_B = \max \mathcal{A} + \varepsilon. \tag{6}$$

For each bin, we compute its number of entries c_b and width w_b as

$$c_b = \left|\{m : a_m \in [b_b, b_{b+1})\}\right|, \qquad w_b = b_{b+1} - b_b. \tag{7}$$

We then assign a target sample count α_b to each bin in proportion to w_b/c_b, which corresponds to inverse sampling density, subject to

$$\sum_b \alpha_b = \min(N, |\mathcal{A}|), \qquad \alpha_b \le c_b. \tag{8}$$

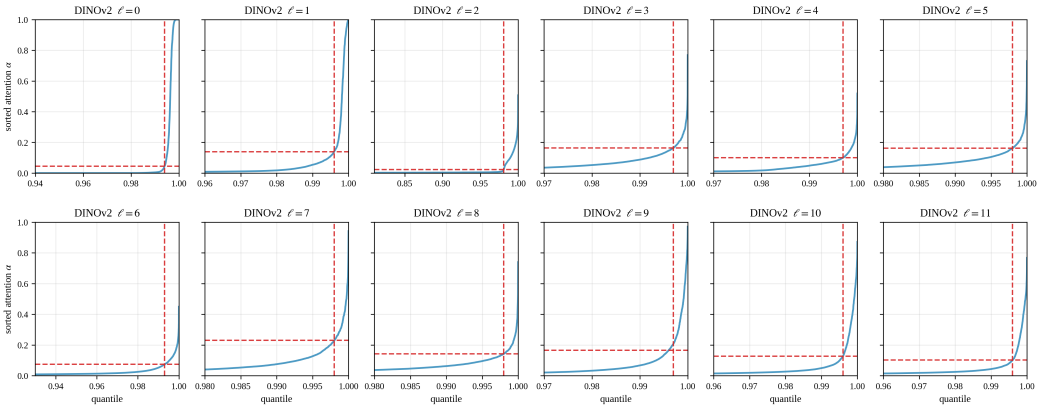


Figure 8: L-method thresholding results for all layers of DINOv2.

Because α_b must be integer-valued, any samples remaining after floor rounding are assigned greedily to the bins with the largest residual fractional allocations. Within each bin, α_b entries are sampled uniformly without replacement.

This procedure yields a sample set that is approximately uniform along the attention-magnitude axis, preventing the pool from being dominated by either the dense "just above τ" region or the sparse high-attention tail.

C.3 Recursive Clustering

Algorithm 1 summarizes the recursive clustering procedure. Throughout this subsection, $\mathcal{Z}$ denotes the index set of the QK features assigned to the node currently being processed, and $\mathcal{Z}_{\text{sub}} \subseteq \mathcal{Z}$ the subset actually passed to UMAP and HDBSCAN. The key hyperparameters are as follows:

- **UMAP (McInnes et al., 2018)**: We use 10 output dimensions, cosine distance, $\text{min_dist} = 0$, $\text{random_state} = 0$, and `low_memory=True`. The neighborhood size is set to $\min(50, \max(10, |\mathcal{Z}_{\text{sub}}|/80))$, which reaches $\text{n_neighbors} = 50$ for every node large enough to be split.
- **HDBSCAN (Campello et al., 2013)**: We use `cluster_selection_method="eom"`, `min_samples=15`, and Euclidean distance on the UMAP embedding. We sweep $\text{min_cluster_size} = \max(100, f\,|\mathcal{Z}_{\text{sub}}|)$ over $f \in \{0.30, 0.20, 0.12, 0.08, 0.05, 0.03\}$. The fraction is defined relative to the clustered set $\mathcal{Z}_{\text{sub}}$, which equals $\mathcal{Z}$ unless the large-cluster speedup described below is applied.
- **Silhouette gate**: We set $s_{\text{gate}} = 0.5$. The silhouette score is computed using Euclidean distance on the UMAP embedding and at most $\min(4000, |\text{non-noise}|)$ samples for efficiency.
- **Node budget**: We set $N_{\min} = 2{,}000$, so a node is considered for splitting only when $|\mathcal{Z}| \geq 2N_{\min} = 4{,}000$. The maximum tree depth is $d_{\max} = 3$. In practice, this depth is close to the limit imposed by the sample budget: across the four backbones, 71% of depth-3 nodes contain fewer than 4,000 QK features and therefore could not be split further.
- **Large-cluster speedup**: When $|\mathcal{Z}| > 10{,}000$, we uniformly subsample 10,000 points to form $\mathcal{Z}_{\text{sub}}$ and run UMAP and HDBSCAN on this subset. HDBSCAN noise points in the subset are reassigned to their nearest non-noise neighbors in UMAP space. The resulting labels are then propagated to the full cluster by nearest-neighbor assignment in the original $2D_h$-dimensional feature space.
- **Branching cap**: We impose no branching cap for ViT-B/16 or CLIP ViT-B/16. For the register-augmented DINOv2-B/14 and DINOv3 ViT-B/16 backbones, we set $k_{\max} = 4$. Without this constraint, HDBSCAN can produce up to 26 children at the root, resulting in trees that are too shallow for useful hierarchical interpretation.

Algorithm 1 Recursive clustering for a single head

Require: features $\{\mathbf{z}_n\}_{n=1}^{N}$, depth d, $d_{\max}$, $N_{\min}$, s_{gate}, $k_{\max}$ (optional)

function BUILDNODE($\mathcal{Z}, d$)
  if $|\mathcal{Z}| < 2N_{\min}$ **or** $d \geq d_{\max}$ **then**
    return leaf $\mathcal{Z}$
  end if
  $\mathcal{Z}_{\text{sub}} \leftarrow \mathcal{Z}$ if $|\mathcal{Z}| \leq 10{,}000$, otherwise a uniform 10,000-point subsample of $\mathcal{Z}$
  $U \leftarrow \text{UMAP}_{10}(\{\mathbf{z}_n\}_{n \in \mathcal{Z}_{\text{sub}}})$
  for $f \in \{0.30, 0.20, 0.12, 0.08, 0.05, 0.03\}$ **do**
    $L_f \leftarrow \text{HDBSCAN}(U;\ \text{mcs} = \max(100,\ f|\mathcal{Z}_{\text{sub}}|))$
    if $k(L_f) \geq 2$ **and** ($k_{\max}$ undefined **or** $k(L_f) \leq k_{\max}$) **then**
      store $(L_f,\ \text{sil}(U, L_f))$
    end if
  end for
  if no candidate is stored **then**
    return leaf $\mathcal{Z}$
  end if
  $L^\star \leftarrow \arg\max$ over stored candidates by silhouette
  if $\text{sil}^\star < s_{\text{gate}}$ **then**
    return leaf $\mathcal{Z}$
  end if
  Reassign HDBSCAN noise points in $L^\star$ to their nearest non-noise neighbors in U
  if $\mathcal{Z}_{\text{sub}} \neq \mathcal{Z}$ **then**
    Propagate $L^\star$ to $\mathcal{Z}$ by nearest-neighbor assignment in $\mathbb{R}^{2D_h}$
  end if
  Partition $\mathcal{Z}$ according to $L^\star$ into children $\mathcal{Z}_1, \ldots, \mathcal{Z}_k$
  return internal node with children $\{\text{BUILDNODE}(\mathcal{Z}_c, d{+}1)\}$
end function

C.4 PER-NODE CLASSIFIERS

At each internal node, we fit a random-forest classifier (Breiman, 2001) with 30 trees, `max_depth=8`, and a fixed random seed. The classifier is trained on the node-specific feature set $\{\mathbf{z}_n\}_{n \in \mathcal{Z}}$, using the HDBSCAN-assigned child cluster as the target label.

At test time, prediction proceeds top-down from the root. At each internal node, the corresponding classifier assigns the input feature to one of its child clusters, after which traversal continues through the predicted child. Because HDBSCAN cluster labels need not be contiguous, we explicitly maintain the mapping from cluster labels to child nodes. The procedure continues until a leaf node is reached.

This lightweight configuration is sufficient in practice because each classifier only needs to distinguish among a small number of sibling clusters that have already been verified to be well separated by the silhouette-based splitting criterion.

C.5 FORMAL DEFINITIONS FOR CLUSTER CONSISTENCY

This section provides the formal definitions of the semantic-agreement statistics introduced in §5.2 and states the notation used there. These include the probe embeddings $\mathbf{e}_I$, the per-image weighted mean direction $\bar{\mathbf{e}}^{\text{Q}}_{I,v}$, and the cluster-consistency statistic $\bar{r}^{\text{Q}}_v$. We describe the Q side. The K side is defined analogously, with h^{K} (resp. $\tilde{h}^{\text{K}}$) replacing h^{Q} (resp. $\tilde{h}^{\text{Q}}$) and sums taken over the query axis instead of the key axis.

Heatmaps h^{Q} and $\tilde{h}^{\text{Q}}$. Let $T = 1 + n_{\text{reg}} + P^2$ denote the number of tokens per image, consisting of one CLS token, n_{reg} register tokens, and P^2 image-patch tokens, where $n_{\text{reg}} \in \{0, 4\}$ depends on the backbone. We index all tokens by $i, j \in \{0, 1, \ldots, T-1\}$, with image-patch indices given by $\{1 + n_{\text{reg}}, \ldots, T-1\}$. Let $\alpha_I^{\ell,h} \in \mathbb{R}^{T \times T}$ denote the row-stochastic attention matrix of head (ℓ, h) on image I, recomputed offline from the cached per-head $\mathbf{Q}$ and $\mathbf{K}$ as $\alpha_I^{\ell,h}(i,j) = \text{softmax}_j(\mathbf{q}_i^{I\top}\mathbf{k}_j^I/\sqrt{d_h})$, with no additional masking. Let $\text{mem}_v(i,j) \in \{0,1\}$ indicate whether the path classifiers of $\mathcal{T}^{\ell,h}$

(§5.1) route the pair $(\mathbf{q}_i^I, \mathbf{k}_j^I)$ to node v. The two heatmaps at query position i are defined as

$$h^{\mathrm{Q}}_{I,v}(i) \;\coloneqq\; \frac{\sum_{j=0}^{T-1} \alpha_I^{\ell,h}(i,j)\,\mathrm{mem}_v(i,j)}{\sum_{j=0}^{T-1} \alpha_I^{\ell,h}(i,j)}, \qquad \tilde{h}^{\mathrm{Q}}_{I,v}(i) \;\coloneqq\; \frac{1}{T}\sum_{j=0}^{T-1} \mathrm{mem}_v(i,j). \tag{9}$$

Both sums run over all token positions, including CLS and register tokens, rather than only image-patch keys. This matches the implementation, where the path classifiers are applied to the full $T \times T$ grid of (i, j) pairs before any spatial slicing. After computing Equation (9), we retain only image-patch query positions and reshape the result into a $P \times P$ map. CLS and register query positions are excluded from all downstream metrics.

Probe and patch-grid resampling. Let $\mathbf{e}_I(i) \in \mathbb{S}^{D-1}$ denote the ℓ_2-normalized probe-model embedding at image-patch position i of image I, where $D = 768$. Stacking these embeddings over the $P \times P$ grid yields $\mathbf{e}_I \in \mathbb{R}^{P\times P\times D}$, as used in §5.2. For the ViT, CLIP, and DINOv2 backbones, we use frozen DINOv3 ViT-B/16 (Siméoni et al., 2025) as the probe. For the DINOv3 backbone, we instead use frozen DINOv2-B/14 (Oquab et al., 2024). Thus, the probe model is external to the dissected backbone in every case. When the probe and the dissected backbone use different patch grids, we bilinearly resample the probe's pre-normalization patch tokens from its native $P_{\mathrm{probe}} \times P_{\mathrm{probe}}$ grid to the backbone's $P \times P$ grid before computing $\mathbf{e}_I(i)$. Here, $P_{\mathrm{probe}} = 14$ for DINOv3 and $P_{\mathrm{probe}} = 16$ for DINOv2, while $P = 14$ for ViT, CLIP, and DINOv3 and $P = 16$ for DINOv2.

Per-image weighted mean direction. For each candidate image I and chain node v, we define the heatmap-weighted mean direction in probe space as

$$\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v} \;\coloneqq\; \frac{\sum_i h^{\mathrm{Q}}_{I,v}(i)\,\mathbf{e}_I(i)}{\sum_i h^{\mathrm{Q}}_{I,v}(i)} \;\in\; \mathbb{R}^D, \qquad \left\|\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}\right\|_2 \le 1. \tag{10}$$

The sums run over the P^2 image-patch positions. Because each $\mathbf{e}_I(i)$ is ℓ_2-normalized, $\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}$ lies in the unit ball and represents the heatmap-weighted mean of the highlighted patch embeddings. Its norm equals one only when all patches with positive h^{Q} have the same probe direction.

If $\sum_i h^{\mathrm{Q}}_{I,v}(i) = 0$, then $\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}$ is undefined and the corresponding image is excluded from subsequent aggregation. This case occurs when node v has no membership pairs on image I, or, under the *highonly* regime of §E.2, when no surviving attention pair originates from an image-patch query. The same definition applies with $\tilde{h}^{\mathrm{Q}}$ in place of h^{Q}.

Prototype-image selection. We use a two-stage selection procedure that mirrors the role card construction. **Stage 1 (candidate pool, α-independent).** For each chain node v, let $\mathcal{P}_v$ denote the $N_{\mathrm{cand}} = 50$ images that contribute the largest numbers of $(\mathbf{q}, \mathbf{k})$ pairs to v in the training pool used to construct $\mathcal{T}^{\ell,h}$. Concretely, we count the source-image frequencies of the QK features $\{\mathbf{z}_n\}_{n\in v}$ and retain the top 50 images. Because $\mathcal{P}_v$ depends only on the cluster assignments of the QK features, the same candidate pool is used for both heatmap definitions (h and $\tilde{h}$) and for both the all and highonly regimes. **Stage 2 (top k ranking, mode-dependent).** The Q-prototype set $\mathcal{Q}_k(v)$ is defined as the top k images in $\mathcal{P}_v$, ranked by $\max_i h^{\mathrm{Q}}_{I,v}(i)$, where the maximum is taken over the $P \times P$ image-patch heatmap. The K-prototype set $\mathcal{K}_k(v)$ is defined symmetrically using $h^{\mathrm{K}}_{I,v}$. For the uniform-α variant, Stage 2 instead ranks images using $\max_i \tilde{h}^{\mathrm{Q}}_{I,v}(i)$ (resp. $\tilde{h}^{\mathrm{K}}$). Thus, each heatmap definition induces its own prototype ranking over the shared candidate pool $\mathcal{P}_v$. We use $k = 10$ in the main figure and $k = 50 = N_{\mathrm{cand}}$ as a robustness check (§F.2). When $k = 50$, Stage 2 retains the entire candidate pool and therefore introduces no additional ranking-based selection.

Cluster-consistency statistic. Let $\mathcal{Q}^{\circ}_k(v) \subseteq \mathcal{Q}_k(v)$ denote the Q-prototype images for which $\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}$ is defined, that is, those with $\sum_i h^{\mathrm{Q}}_{I,v}(i) > 0$. We aggregate their per-image vectors using the norm of the cross-image mean:

$$\bar{r}^{\mathrm{Q}}_v \;\coloneqq\; \left\| \frac{1}{|\mathcal{Q}^{\circ}_k(v)|} \sum_{I\in\mathcal{Q}^{\circ}_k(v)} \bar{\mathbf{e}}^{\mathrm{Q}}_{I,v} \right\|_2 \;\in\; [0,1]. \tag{11}$$

A high $\bar{r}^{\mathrm{Q}}_v$ requires two forms of consistency. First, each per-image vector $\bar{\mathbf{e}}^{\mathrm{Q}}_{I,v}$ must have a large norm, indicating that the highlighted patches are concentrated in probe space. Second, these vectors must be similarly oriented across prototype images, indicating cross-image semantic alignment. If

different prototype images highlight regions whose probe-space directions disagree, their vectors cancel in the cross-image average, and $\bar{r}_v^{\mathrm{Q}}$ remains small even when each individual norm $\|\bar{\mathbf{e}}_{I,v}^{\mathrm{Q}}\|_2$ is close to 1. This construction is closely related to the resultant-length statistic used in directional statistics (Mardia & Jupp, 2000; Banerjee et al., 2005), with the difference that the per-image vectors here lie in the unit ball rather than being individually normalized to the unit hypersphere. The $[0, 1]$ bound follows directly from $\|\bar{\mathbf{e}}_{I,v}^{\mathrm{Q}}\|_2 \leq 1$. We use the norm of the cross-image mean rather than the alternative mean of per-image norms,

$$\frac{1}{|\mathcal{Q}_k^\circ(v)|} \sum_{I \in \mathcal{Q}_k^\circ(v)} \left\| \bar{\mathbf{e}}_{I,v}^{\mathrm{Q}} \right\|_2,$$

since the latter is invariant to the relative orientation of the per-image vectors and therefore cannot detect disagreement across prototype images. The remaining variants are defined analogously by replacing h^{Q} with h^{K}, $\tilde{h}^{\mathrm{Q}}$, or $\tilde{h}^{\mathrm{K}}$, as appropriate.

Parent-matched null. A high value of $\bar{r}_v^{\mathrm{Q}}$ may arise partly from the choice of prototype images rather than from node v itself. To control for this effect, let $u \coloneqq \mathrm{parent}(v)$ in $\mathcal{T}^{\ell,h}$. We reuse the same ranked prototype set $\mathcal{Q}_k(v)$, evaluate Equation (10) with the parent heatmap $h_{I,u}^{\mathrm{Q}}$ in place of $h_{I,v}^{\mathrm{Q}}$, and aggregate the resulting vectors via Equation (11). We denote the result by $\bar{r}_{u,\mathcal{Q}_k(v)}^{\mathrm{Q}}$. Since $\mathrm{mem}_v(i,j) \leq \mathrm{mem}_u(i,j)$ pointwise, the parent's heatmap covers the region assigned to v together with those assigned to its sibling nodes. The same inequality implies that the images retained under u form a superset of $\mathcal{Q}_k^\circ(v)$, so the null is never averaged over fewer images than the real statistic. The comparison therefore measures whether the additional specificity of v yields greater semantic consistency, while holding the image ranking, the evaluation regime, and the heatmap definition (h vs. $\tilde{h}$) fixed. If v captures finer semantic structure than its parent, we expect $\bar{r}_v^{\mathrm{Q}} > \bar{r}_{u,\mathcal{Q}_k(v)}^{\mathrm{Q}}$. The depth-0 root is used only as the parent-matched null for depth-1 nodes and is not itself included as an evaluated node.

Chain restriction. For each head (ℓ, h), we evaluate the metrics only on nodes belonging to root-to-depth-3 lineages. Specifically, a node v is included if $\mathrm{depth}(v) \in \{1, 2, 3\}$ and it either has a depth-3 descendant or is itself at depth 3. Heads whose trees contain no depth-3 node are excluded entirely. This leaves 312 chain-restricted heads pooled across the four backbones. This restriction removes branches that terminate at depth 1 or 2, ensuring that the depth-progression analysis compares nodes drawn from the same set of root-to-depth-3 trajectories at every depth.

Implementation note: order of stages. For each head, Stage 1 first determines $\mathcal{P}_v$ for every chain node v from the cluster pair counts. We then compute and cache $(h_{I,v}^{\mathrm{Q}}, h_{I,v}^{\mathrm{K}})$ for every chain node and every image in $\bigcup_v \mathcal{P}_v$. A single top-down routing pass over the image's query-key pairs provides the membership assignments needed for all nodes along the traversed paths. Stage 2 then ranks prototype images using this cache. The same cached heatmaps are reused for the parent-matched null, since they include both each node and its parent.

D Full Evidence of H1

This section expands §4 in two parts. §D.1 defines the diagnostics used in the main text and describes the Dip-means procedure on which three of them depend. §D.2 then reports the full results, including standard deviations and additional metrics. For completeness, we additionally report DINOv2 results, which are omitted from Table 1 in the main text.

D.1 Metrics and the Dip-means Procedure

Recall that $\{\mathbf{z}_n\}_{n=1}^N \subset \mathbb{R}^{2D_h}$ denotes the set of joint query-key features for a single head after ℓ_2-normalization, sampled from the L-method-thresholded pool. Below, we define the metrics used in §4 and Tables 3–4.

Silhouette coefficient (Rousseeuw, 1987). For each point $\mathbf{z}_n$ assigned to a cluster, let $a(n)$ denote the mean distance from $\mathbf{z}_n$ to the other points in the same cluster and $b(n)$ the mean distance to points in the nearest neighboring cluster. The silhouette score is then defined as $s(n) = (b(n) -$

$a(n))/\max(a(n), b(n)) \in [-1, 1]$, and the reported value is the average of $s(n)$ over all non-noise points. As a rule of thumb, $s \geq 0.7$ indicates strong cluster separation, $0.5 \leq s < 0.7$ moderate separation, and $s \approx 0$ little or no cluster structure.

HDBSCAN noise % (Campello et al., 2013). This metric is the fraction of features that HDBSCAN leaves unassigned. Low values indicate that a large fraction of points can be assigned to coherent clusters, whereas high values indicate that much of the data does not belong to any sufficiently dense cluster. The noise fraction therefore provides a complementary view to the silhouette score by accounting for points that HDBSCAN rejects.

Hopkins statistic (Banerjee & Davé, 2004). The Hopkins statistic compares nearest-neighbor distances among the observed features $\mathbf{z}_n$ with those obtained from reference points sampled uniformly from the bounding box of the data. Values lie in $[0, 1]$, with values near 0.5 indicating little cluster tendency and values approaching 1 indicating stronger departure from spatial randomness. A common rule of thumb treats $H > 0.75$ as evidence of cluster tendency.

Dip-means procedure (Kalogeratos & Likas, 2012). The remaining three diagnostics are derived from Dip-means, a recursive clustering procedure whose splitting criterion is based on Hartigan's dip test of unimodality (Hartigan & Hartigan, 1985). Given a cluster $\mathcal{C} \subseteq \{\mathbf{z}_n\}$, the procedure samples a set of *viewers* $\mathcal{V} \subseteq \mathcal{C}$. For each viewer $\mathbf{z}' \in \mathcal{V}$, it constructs the one-dimensional distribution

$$\mathcal{D}_{\mathbf{z}'} \coloneqq \{\|\mathbf{z} - \mathbf{z}'\|_2 : \mathbf{z} \in \mathcal{C} \setminus \{\mathbf{z}'\}\}. \tag{12}$$

Hartigan's dip test is then applied to $\mathcal{D}_{\mathbf{z}'}$ at significance level α. A viewer is said to *vote for splitting* when unimodality is rejected. The cluster $\mathcal{C}$ is split using 2-means when the fraction of splitting votes exceeds v_{thd}, and otherwise becomes a leaf. This procedure is applied recursively to the resulting child clusters, yielding a binary tree whose depth reflects the maximum number of recursively accepted splits along a root-to-leaf path.

Dip-means configuration. We run Dip-means on a uniform $N = 2{,}000$-pair subsample of the $10{,}000$-pair pool used for the flat-clustering diagnostics in §4. We use this smaller subsample for computational efficiency, as the recursive procedure repeatedly computes pairwise distances within each cluster. This subsampling reduces the per-head runtime from hours to minutes in our implementation. We set $\alpha = 0.05$ and $v_{\text{thd}} = 0.01$ following Kalogeratos & Likas (2012), and sample $|\mathcal{V}| = 30$ viewers uniformly from each cluster. We additionally use a minimum cluster size of 50 and a maximum recursion depth of 8.

Dip-means tree depth. This metric records the maximum depth of the binary split tree returned by the procedure above. Depth 0 indicates that no split was accepted at the root cluster. Depth $d \geq 1$ indicates d nested levels of accepted splits.

Number of leaves. This metric counts the number of terminal clusters in the Dip-means tree. A value of 1 indicates that no splits occurred, while larger values correspond to finer recursive partitioning.

Splittable %. This metric is the fraction of the 144 heads in a backbone whose Dip-means tree reaches depth ≥ 1, that is, the fraction of heads for which Dip-means accepts at least one split. It therefore measures how frequently multimodal structure is detected across attention heads.

D.2 Full Results

Flat clustering diagnostics. Table 3 reports the silhouette coefficient, HDBSCAN noise %, and Hopkins statistic across the four backbones and four reference conditions, with standard deviations computed over the 144 heads of each backbone. The real data yield approximately $3\,\%$ noise, whereas the *random* null yields $100\,\%$. Hopkins, in contrast, has limited discriminative value in this setting. It is approximately 1.00 for both the real and *shuffled* data, and remains as high as 0.93 even for the *random* null. This illustrates that the Hopkins statistic can saturate in this high-dimensional setting, limiting its discriminative value.

Table 3: Flat clustering diagnostics on $\{\mathbf{z}_n\}$ ($N = 10{,}000$ query-key pairs per head sampled from the L-method-thresholded, bin-balanced pool of §4; 144 heads per backbone; mean ± standard deviation).

Backbone	Metric	real	shuffled	cov-matched	dim-permuted	random
ViT	Silhouette (↑)	0.80 ± 0.13	0.80 ± 0.13	0.55 ± 0.20	0.16 ± 0.14	0.00
	Noise % (↓)	2.6	2.8	54.7	64.4	100.0
	Hopkins (↑)	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.01	0.95 ± 0.03	0.93
CLIP	Silhouette (↑)	0.81 ± 0.15	0.80 ± 0.15	0.60 ± 0.24	0.12 ± 0.16	0.00
	Noise % (↓)	3.5	4.6	49.9	81.8	100.0
	Hopkins (↑)	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.02	0.94 ± 0.03	0.93
DINOv2	Silhouette (↑)	0.81 ± 0.12	0.81 ± 0.12	0.57 ± 0.20	0.17 ± 0.13	0.00
	Noise % (↓)	3.6	3.4	57.6	59.7	100.0
	Hopkins (↑)	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.02	0.97 ± 0.03	0.93
DINOv3	Silhouette (↑)	0.83 ± 0.16	0.81 ± 0.13	0.56 ± 0.16	0.18 ± 0.15	0.00
	Noise % (↓)	4.3	6.3	53.9	60.2	100.0
	Hopkins (↑)	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.01	0.96 ± 0.03	0.93

Dip-means diagnostics. Table 4 reports tree depth, number of leaves, and Splittable % for the same backbones and reference conditions. Across backbones, the real data yield average depths between 1.70 and 3.30 and between 3.22 and 5.57 leaves, whereas the cov-matched, dim-permuted, and random conditions remain near 0.00 and 1.00, respectively. The Splittable fraction ranges from 64 % to 86 % on the real data and stays at or below 1 % for these three conditions. The Splittable fraction therefore provides a clear head-level distinction between the real data and these null conditions. These results provide evidence of recursive multimodal structure in the real QK distributions that is largely absent from the three synthetic conditions. The *shuffled* condition retains substantial recursive structure because it preserves the marginal query and key representations. Nevertheless, the real pairing yields consistently higher Splittable fractions and deeper trees across all four backbones.

Table 4: Full Dip-means results across the four backbones using an $N = 2{,}000$-pair subsample from the pool described in §4 ($|\mathcal{V}| = 30$, $\alpha = 0.05$, $v_{\text{thd}} = 0.01$). Values are reported as mean ± standard deviation over the 144 heads of each backbone. *Splittable %* denotes the fraction of heads for which Dip-means accepts at least one split.

Backbone	Metric	real	shuffled	cov-matched	dim-permuted	random
ViT	Tree depth (↑)	2.14 ± 2.01	1.23 ± 1.67	0.01 ± 0.12	0.00	0.00
	Number of leaves (↑)	3.77 ± 4.09	2.56 ± 2.54	1.01 ± 0.12	1.00	1.00
	Splittable % (↑)	64	50	1	0	0
CLIP	Tree depth (↑)	1.70 ± 1.78	1.33 ± 1.52	0.01 ± 0.12	0.00	0.00
	Number of leaves (↑)	3.22 ± 3.47	2.64 ± 2.18	1.01 ± 0.12	1.00	1.00
	Splittable % (↑)	68	63	1	0	0
DINOv2	Tree depth (↑)	2.54 ± 2.21	1.54 ± 1.99	0.01 ± 0.08	0.00	0.00
	Number of leaves (↑)	4.28 ± 3.46	3.14 ± 3.68	1.01 ± 0.08	1.00	1.00
	Splittable % (↑)	68	47	1	0	0
DINOv3	Tree depth (↑)	3.30 ± 1.90	2.70 ± 2.11	0.01 ± 0.08	0.00	0.00
	Number of leaves (↑)	5.57 ± 3.18	5.05 ± 3.74	1.01 ± 0.08	1.00	1.00
	Splittable % (↑)	86	73	1	0	0

E JUSTIFICATION OF PIPELINE DESIGN CHOICES

This section collects three diagnostics that justify pipeline choices made in §5.1 and §5.2: ① why we construct the QK tree from the L-method-thresholded high-attention pool rather than from the full attention distribution (§E.1), ② why path classifiers trained on this high-attention pool remain informative when applied to all attention pairs at evaluation (§E.2), and ③ whether role cards constructed from all attention pairs are more interpretable than their highonly counterparts (§E.3).

E.1 L-METHOD ABLATION: FULL-RANGE SAMPLING SUBSTANTIALLY WEAKENS CLUSTERING

A natural alternative to L-method elbow filtering is to sample $50{,}000$ pairs uniformly across 10 equal-count quantile bins of the *full* attention distribution, so that low- and high-attention regions are both represented. We re-ran the entire pipeline (identical UMAP→HDBSCAN, $s_{\text{gate}} = 0.5$, $d_{\max} = 3$, $N_{\min} = 2{,}000$) with this alternative sampling on all 144 ViT-B/16 heads. Table 5 reports the result.

Table 5: L-method elbow filtering versus uniform-quantile sampling across the full attention distribution. *Splittable heads* denotes the number of heads out of 12 per layer for which at least one HDBSCAN split survives the silhouette gate. *Mean nodes* denotes the mean number of tree nodes per head, including leaves.

	Splittable heads		Mean nodes	
Layer	L-method	full-range	L-method	full-range
L0	12/12	4/12	12.8	4.0
L1	11/12	4/12	14.2	3.8
L2	12/12	3/12	29.7	5.6
L3	11/12	4/12	21.0	4.9
L4	11/12	5/12	14.8	5.6
L5	12/12	6/12	9.4	3.9
L6	12/12	5/12	12.5	4.3
L7	12/12	3/12	10.2	2.2
L8	12/12	0/12	9.8	1.0
L9	11/12	0/12	9.1	1.0
L10	12/12	2/12	10.3	2.2
L11	8/12	2/12	10.5	1.9
All	**136/144**	**38/144**	**13.7**	**3.4**

L-method filtering yields multi-cluster trees in 94% of heads (mean 13.7 nodes per head), whereas full-range stratified sampling collapses 74% of heads to a single root cluster (mean 3.4 nodes per head). In particular, L8 and L9 collapse entirely under full-range sampling. These results indicate that the large population of low-attention pairs substantially weakens the cluster structure recovered by our pipeline. When such pairs are included throughout the sampled attention range, HDBSCAN much less frequently identifies partitions that satisfy the silhouette criterion. The L-method threshold therefore serves not only as a computational filter, but also as a practical mechanism for focusing the clustering procedure on the high-attention region where recoverable QK structure is more pronounced.

E.2 GENERALIZATION FROM HIGHONLY TRAINING TO ALL-PAIR EVALUATION

Train/test mismatch. The QK tree in §5.1 is constructed from the L-method-thresholded high-attention pool ($N = 50{,}000$ per head, all satisfying $\alpha^{\ell,h} \geq \tau^{\ell,h}$). In contrast, the heatmaps $h^{\mathrm{Q}}_{I,v}$ and $h^{\mathrm{K}}_{I,v}$ in §6.1, and therefore the consistency analysis in §5.2, evaluate the path classifiers on *all* query-key pairs of an image, including pairs whose attention values fall below $\tau^{\ell,h}$. We directly verify that the observed depth trend in $\bar{r}^s$ persists despite this train/test mismatch.

Test setup. For every chain-restricted node, we recompute the full pipeline of §5.2 under a *highonly* regime. When constructing $h^{\mathrm{Q}}_{I,v}$ (§6.1), we zero out all attention entries satisfying $\alpha_I(i,j) < \tau^{\ell,h}$ in both the numerator and denominator, thereby restricting the heatmap to the same above-threshold attention pool used to construct the QK tree. Prototype-image selection, the parent-matched null, and all remaining components of the pipeline are left unchanged. We compare these *highonly-regime* measurements against the *all-regime* measurements reported in the main figure, which use unthresholded heatmaps and therefore evaluate the path classifiers on every query-key pair.

Result. Table 6 reports the pooled real-versus-null gap across the 312 chain-restricted heads from the four backbones. The gap is positive in every cell under both regimes. At depth 3, the all-regime gap reaches $+0.043$ on the Q side and $+0.045$ on the K side, compared with $+0.039$ and $+0.044$ under the highonly regime. On both sides and in both regimes, $\bar{r}^s$ increases monotonically with depth.

Table 6: Highonly vs. all-regime semantic agreement, pooled across the 312 chain-restricted heads of the four backbones ($k = 10$). *r gap* = $\bar{r}^s_v - \bar{r}^s_u$ (positive = v has higher semantic agreement than its parent u); n = number of contributing chain nodes.

		all-regime				**highonly regime**			
Depth	Side	$\bar{r}^s_v$	$\bar{r}^s_u$	gap	n	$\bar{r}^s_v$	$\bar{r}^s_u$	gap	n
1	Q	0.765	0.747	+0.018	410	0.787	0.766	+0.020	409
1	K	0.808	0.760	+0.049	297	0.849	0.820	+0.029	251
2	Q	0.790	0.766	+0.024	522	0.809	0.787	+0.023	521
2	K	0.828	0.801	+0.027	351	0.862	0.842	+0.019	282
3	Q	0.829	0.786	+0.043	1615	0.850	0.811	+0.039	1595
3	K	0.870	0.825	+0.045	1111	0.905	0.861	+0.044	917

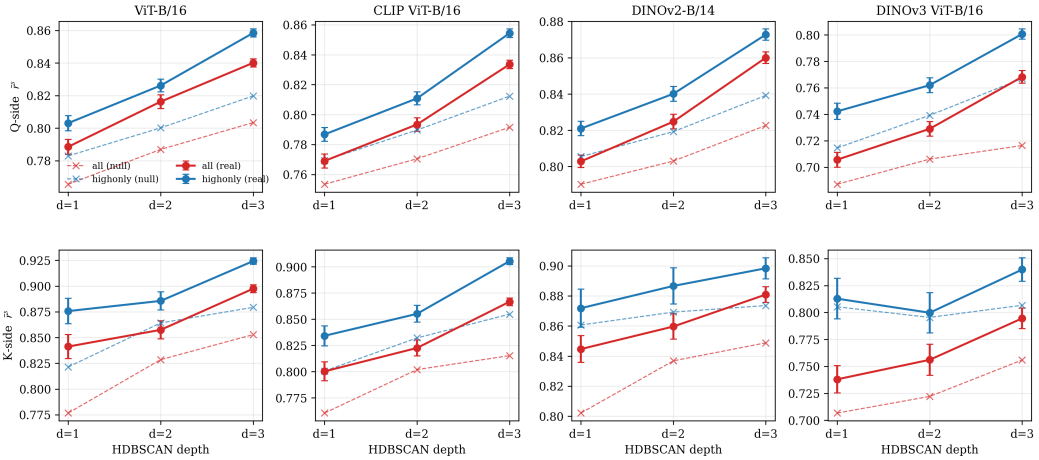


Figure 9: Semantic agreement $\bar{r}^s$ across QK tree depth, all-regime (red) vs. highonly regime (blue), pooled within each backbone (columns) and side (rows). Solid lines are the real values $\bar{r}^s_v$; dashed lines are the parent-matched null $\bar{r}^s_u$. Both regimes show a monotonic depth trend and real-vs-null separation in every cell. Error bars are SEMs across the 312 chain-restricted heads, with K-side highonly node counts slightly reduced for sink-dominated heads (see text).

Figure 9 visualizes these trends across the eight (side, backbone) panels. The absolute highonly values of $\bar{r}^s$ are consistently slightly larger than the corresponding all-regime values. This is consistent with the fact that restricting the heatmap to above-threshold pairs produces sparser and more concentrated activations. Nevertheless, both the depth trend and the strict dominance over the parent-matched null remain unchanged. The K-side sample counts in the highonly regime are slightly lower than those in the all regime (251 vs. 297 at depth 1, and 917 vs. 1,111 at depth 3). This reduction occurs because some heads concentrate nearly all attention mass on a single sink token, typically CLS. For some image-patch key positions j in such heads, all incoming attention values $\alpha_I(i, j)$ fall below $\tau^{\ell,h}$, causing the corresponding highonly K-side heatmap to vanish. A node drops out of the K-side highonly average when this happens on all of its prototype images, while the same node still contributes in the all regime. This accounts for most of the count discrepancy. Overall, the real-versus-null gap remains positive at every depth and on both sides under both regimes, indicating that the depth trend is robust to applying the path classifiers beyond the high-attention pool on which they were trained.

E.3 Human Preference Study on Role-Card Interpretability

We evaluate whether participants find role cards constructed under the *all* regime more helpful for interpretation than their *highonly* counterparts.

Procedure. We conducted a pairwise preference study with three volunteer participants from our laboratory who were familiar with the task domain. None of the participants were authors of this paper. Participants evaluated a fixed set of 127 paired role-card images sampled across layers and backbones. Each pair contained the *all* and *highonly* variants for the same head and node. Participants selected *first*, *second*, or *tie* according to which visualization was more helpful for understanding the node's functional behavior. The trial order was independently randomized for each participant, and the left-right placement of the *all* variant was randomized within each trial. Participants completed the study independently and did not discuss their responses. Following the pairwise preference scoring protocol of Li et al. (2023), we assign a score of 1 to an *all* preference, 0 to a *highonly* preference, and 0.5 to a tie. Examples of the compared role-card variants are shown in Figures 11 and 12. The trial interface is shown in Figure 10.

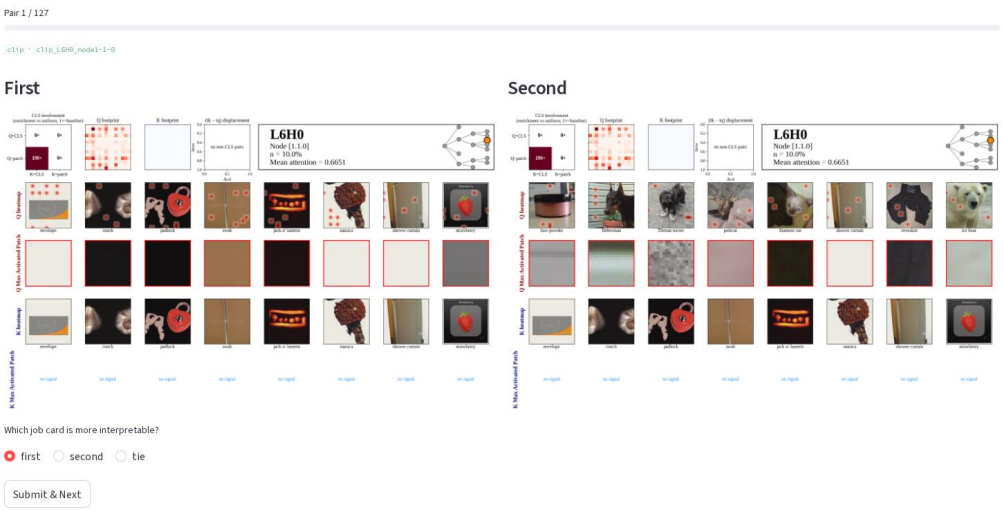


Figure 10: Trial interface shown to participants. The two cards are presented side-by-side as "First" and "Second." Participants select *first*, *second*, or *tie* according to which visualization is more helpful for interpretation, and then advance with the Submit & Next button.

Table 7: Human-preference results. Counts are pooled by backbone over the three response categories *all*, *highonly*, and *tie*. The score is the preference score of the *all* variant, with ties counted as half a preference.

Backbone	Trials	*all*	*highonly*	Tie	Score
CLIP	138	23	11	104	0.543
DINOv2	114	27	8	79	0.583
ViT	129	35	20	74	0.558
All backbones	381	85	39	257	0.560

Results. Table 7 summarizes the results. Across all backbones, the *all* regime achieved a preference score of 0.560. The corresponding scores were 0.543 for CLIP, 0.583 for DINOv2, and 0.558 for ViT. Most responses were ties, accounting for 257/381 (67.5%) of all judgments, indicating that the two variants were often similarly interpretable to participants. Across the 127 image pairs evaluated by all three participants, 24 pairs (18.9%) received opposing non-tie preferences from different participants. Overall, the study suggests a modest aggregate preference for using all attention pairs when constructing role-card heatmaps.

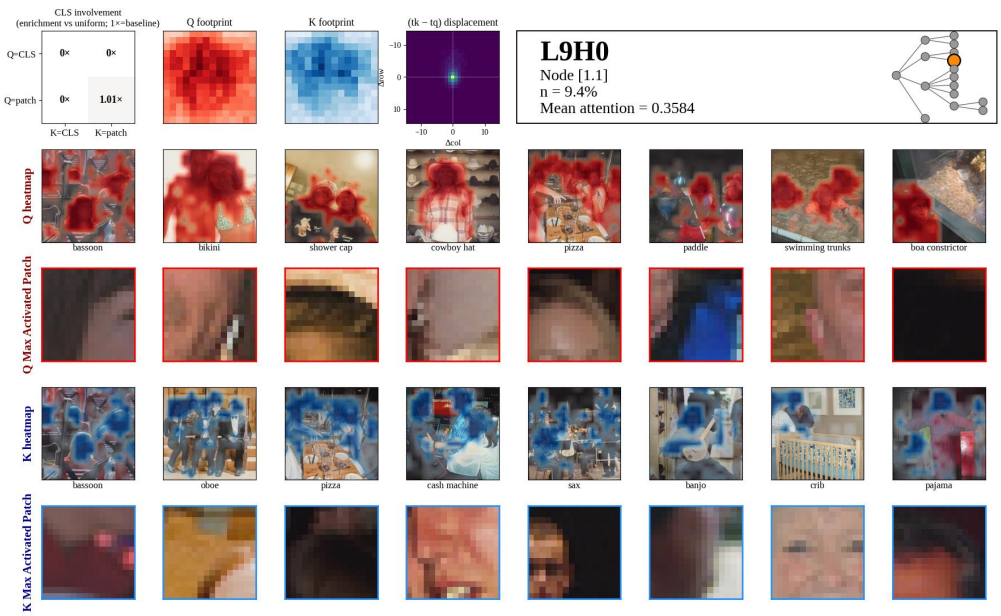


Figure 11: Example role card for the *all* regime. Each trial paired an *all* card with the corresponding *highonly* card for the same head and node.



Figure 12: Example role card for the *highonly* regime corresponding to the same head and node shown in Figure 11. Participants compared the two cards side-by-side.

F Supplementary Cluster-Consistency Analyses

This section complements §5.2 along two axes that are useful for stress-testing the main claim but distract from the main flow: a layer-grouped breakdown that asks whether the depth trend is concentrated in a particular layer range (§F.1) and a robustness check at a much larger prototype count $k = 50$ (§F.2). Both analyses use the same 312 chain-restricted heads and the same parent-matched null as the main figure.

F.1 Layer-Grouped Trends

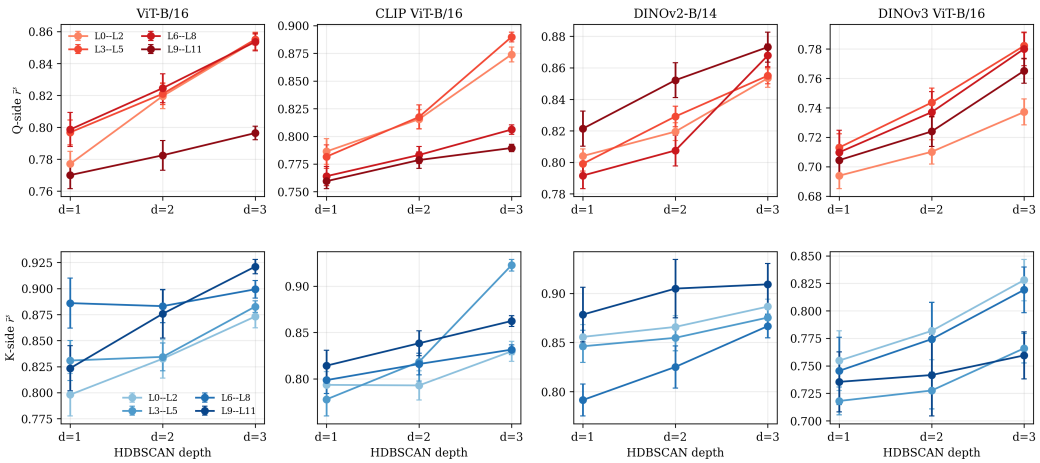


Figure 13: Semantic agreement $\bar{r}^s$ broken down by transformer-layer group (L0–L2, L3–L5, L6–L8, L9–L11, light to dark) for Q (top row, red) and K (bottom row, blue) across the four backbones (columns). $k = 10$. Parent-matched null curves are suppressed for clarity (see Figure 3 for the pooled null pattern). The depth trend is present in every layer group and every backbone, with similar magnitude across layer bands.

The pooled view of Figure 3 averages over all transformer layers, which can mask the possibility that the depth trend is driven entirely by a particular layer range (e.g., that early layers do not subdivide meaningfully and only late-layer heads contribute to the slope). To check this we partition the 12 layers into four equal-sized groups: early (L0–L2), early-mid (L3–L5), late-mid (L6–L8), and late (L9–L11). We then re-pool the same chain-restricted heads within each group. Figure 13 shows the result for semantic agreement, with the four layer groups overlaid in shaded color (light to dark for early to late layers). We suppress the parent-matched null lines for clarity, since each per-cell null pattern matches the pooled null already shown in §5.2.

Two observations are consistent with the pooled main figure. First, the depth trend is present in every layer group and on both Q and K sides for all four backbones: $\bar{r}^s$ increases from depth 1 to depth 3 in nearly every layer-group curve, ruling out the possibility that the main effect comes from a specific layer band. Second, the magnitude of the trend is similar across layer groups, with no single band dominating.

F.2 Robustness to Prototype Count k

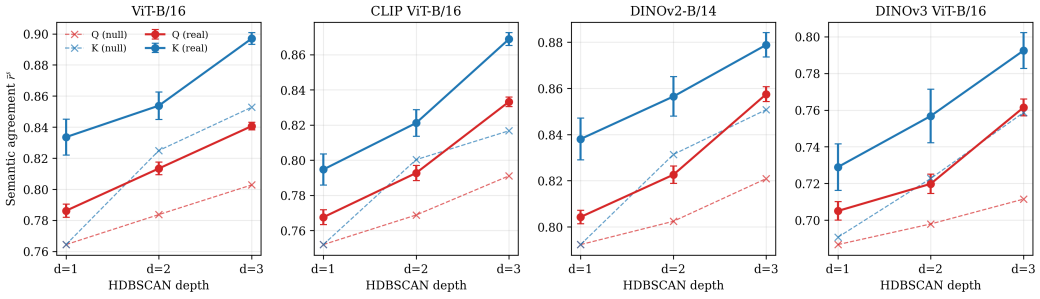


Figure 14: Semantic agreement $\bar{r}^s$ at $k = 50$ prototype images per node (cf. Figure 3 for $k = 10$). The depth trend and the strict null dominance both replicate at the larger k.

The pooled and layer-grouped analyses above use the top $k = 10$ prototype images per node, matching the role card setting up to a small relaxation (§5.2). To verify that the observed depth trend is not an

artifact of the choice of k, we repeat the entire consistency analysis at $k = 50$. Because $N_{\text{cand}} = 50$, this removes the second-stage $\max_q h^q$ ranking altogether and averages over the full stage-one candidate pool. Consequently, any genuine cluster property must remain detectable even after being diluted by images that contribute more weakly to the cluster.

Figure 14 shows that $\bar{r}^s$ exhibits essentially the same behavior at $k = 50$ as at $k = 10$. Across all four backbones and on both sides, $\bar{r}^s$ increases monotonically with depth and remains strictly above its parent-matched null in all 24 (backbone, side, depth) cells. The absolute magnitude of $\bar{r}^s$ is consistently slightly lower at $k = 50$, which is expected because averaging over a broader and less concentrated candidate pool weakens the within-image concentration signal. Nevertheless, both the depth trend and the real-versus-null separation remain intact.

G Causal Relevance of Nodes

The analysis in §5.2 shows that nodes activate on semantically coherent regions, but does not establish whether the interactions assigned to those nodes affect the model's output. Here, we test this question directly through targeted interventions.

Intervention and controls. For an input image, we define the node mask $M_v(i, j) \coloneqq \text{mem}_v(\mathbf{q}_i, \mathbf{k}_j)$ and set the pre-softmax logit of every cell satisfying $M_v(i, j) = 1$ to $-\infty$ (Geva et al., 2023). The softmax is then renormalized over the remaining keys, removing the interactions assigned to v without directly masking the other interactions in the head.[4] A node knockout may have a larger effect simply because the node contains many attention cells. We therefore compare each knockout against two controls whose mask sizes are matched separately for every image. The *uniform* control masks the same total number of cells at uniformly random positions. The *row-matched* control additionally matches the number of masked cells within each query row while randomizing their key positions. The latter controls for both the overall intervention size and its distribution across query rows, allowing us to test whether the specific query-key interactions selected by the node matter beyond these factors.

Protocol. We evaluate all depth-1 nodes from layers $\{3, 6, 9\}$ on 500 ImageNet validation images drawn from 500 distinct classes, with one image per class. Routing uses mem_v itself, namely the same classifier chain introduced in §5.1 applied to each image's query and key vectors. The masks are therefore recomputed independently for every image. Each backbone is evaluated through its corresponding 1000-way prediction head: the supervised classifier for ViT, the public linear-classifier head for DINOv2+reg, the zero-shot text classifier obtained by averaging the 80 canonical prompt templates for CLIP, and a linear probe on frozen DINOv3 features with validation top-1 accuracy of 79.6%. For every node, control, and image, we compute the KL divergence between the clean and intervened class distributions and average it over images. Because the resulting per-node KL values span approximately nine orders of magnitude, from 10^{-9} to 4.3, we summarize the effect using the median across nodes of $\text{KL}_{\text{node}}/\text{KL}_{\text{control}}$. We additionally apply a paired Wilcoxon signed-rank test to the log-ratios. A sign test over the same node pairs gives $p \leq 6 \times 10^{-6}$ for every comparison reported below.

Results. Table 8 summarizes the results. Across all four backbones, node knockout produces a median KL divergence between $2.5\times$ and $13.4\times$ that of the uniform control. Roughly three quarters of nodes produce a larger perturbation than their matched controls, and the effect persists under the stricter row-matched comparison, where the median ratios range from $2.9\times$ to $15.3\times$. These results indicate that node membership identifies query-key interactions with greater functional influence than equally sized random subsets of the same attention matrix. The interventions are also relatively sparse. For ViT, a depth-1 node routes a median of 565 cells per image, corresponding to approximately 1.5% of the attention matrix.

Depth-2 nodes. The depth-1 experiment does not test whether subdivisions below a parent node carry additional functional structure. We therefore repeat the intervention on depth-2 nodes using a

[4] Query rows that would lose all keys are left unmasked, since the softmax of a row whose logits are all $-\infty$ is undefined.

Table 8: Node knockout compared with size-matched controls (depth-1 nodes, layers $3/6/9$, 500 ImageNet validation images). *Ratio* is the median across nodes of $\mathrm{KL}_{\mathrm{node}}/\mathrm{KL}_{\mathrm{control}}$. *Win* is the fraction of nodes for which the node knockout produces a larger KL divergence than its matched control. p is from a paired Wilcoxon signed-rank test on the log-ratios.

		vs. uniform			vs. row-matched		
Backbone	Nodes	Ratio	win	p	Ratio	win	p
ViT (sup.)	102	6.45×	83.3%	1.4×10^{-14}	5.80×	72.5%	4.9×10^{-11}
CLIP	121	2.51×	76.9%	3.7×10^{-13}	2.87×	76.0%	1.2×10^{-12}
DINOv2+reg	87	11.99×	74.7%	2.7×10^{-10}	7.50×	74.7%	6.3×10^{-10}
DINOv3	118	13.42×	78.8%	1.0×10^{-13}	15.29×	77.1%	2.8×10^{-14}

stricter *parent-matched* control, which masks a random subset of the parent's cells with the same cardinality as M_v. If the partition of a parent into its children were arbitrary, the two interventions would be expected to produce similar effects. The node knockout nevertheless produces a larger perturbation on both backbones tested. For ViT, the median ratio is $1.26\times$, with a win rate of 60.7% and $p = 8.3\times10^{-6}$ over 178 nodes. For DINOv2+reg, the corresponding values are $2.31\times$, 75.0%, and $p = 7.4\times10^{-9}$ over 104 nodes. These margins are smaller than those obtained against an unstructured size-matched control. This is expected because a depth-2 node routes substantially fewer cells than a depth-1 node, with medians of 206 and 565 cells per image on ViT, respectively. When restricting the analysis to depth-2 nodes covering at least 0.1% of the attention matrix, the DINOv2+reg effect remains at $1.74\times$ with a win rate of 68.0%. For ViT, the ratio decreases to $1.06\times$ with a win rate of 56.4%, and the corresponding sign test is no longer statistically significant.

H ROLE CARDS

H.1 DETAILS ON EACH PANEL

We give the precise definition of each top-row panel referenced in §6.1. Throughout, v denotes the node being summarized and $\mathcal{P}_v$ its Stage-1 candidate pool of $N_{\text{cand}} = 50$ images, defined in §C.5.

CLS involvement. We classify the query and key of every $(\mathbf{q}, \mathbf{k})$ pair routed to v as CLS, register, or image patch, and tabulate the resulting counts in a 3×3 token-type matrix. For backbones without register tokens, this reduces to a 2×2 matrix with CLS and patch as the row and column labels. Each cell displays the fraction of pairs assigned to v whose query and key belong to the corresponding token types, reported as a percentage to one decimal place.

Q and K footprints. The Q footprint is the per-patch average of $h^{\mathrm{Q}}_{I,v}$ over the candidate pool,

$$\text{footprint}^{\mathrm{Q}}_v(i) \;\coloneqq\; \frac{1}{|\mathcal{P}_v|} \sum_{I \in \mathcal{P}_v} h^{\mathrm{Q}}_{I,v}(i)\,, \tag{13}$$

rendered as a $P\times P$ heatmap on the canonical patch grid. The K footprint is defined analogously using h^{K}. These footprints reveal whether node v consistently activates at particular spatial locations across candidate images or is distributed over a broader region of the query and key grids.

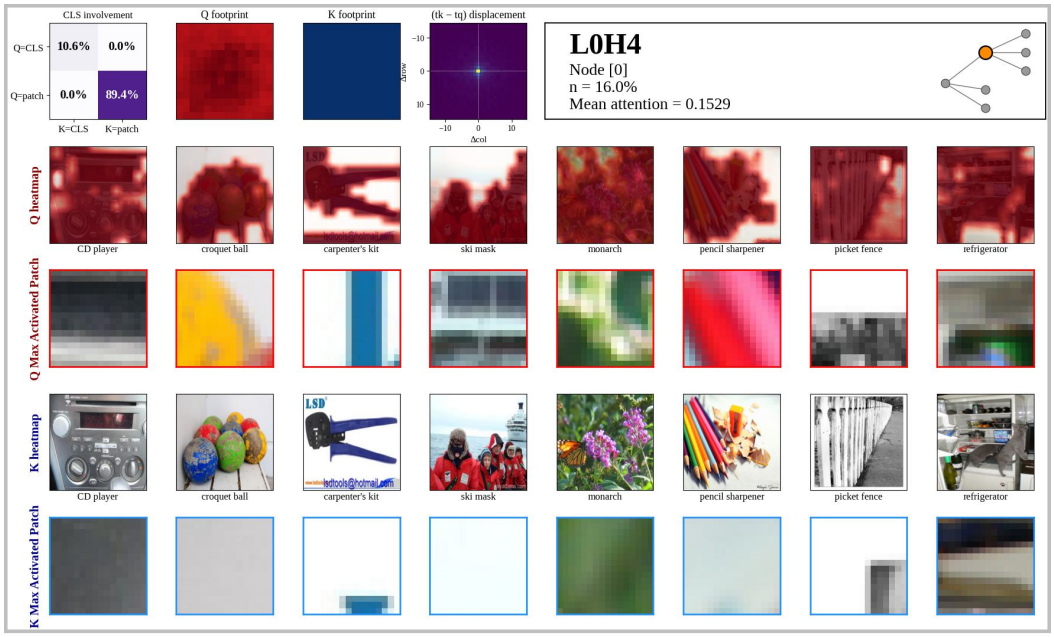
Spatial displacement histogram. For image-patch-to-image-patch pairs routed to v, we compute the relative grid displacement

$$(\Delta\text{row},\, \Delta\text{col}) \;\coloneqq\; (r_{\mathrm{K}} - r_{\mathrm{Q}},\, c_{\mathrm{K}} - c_{\mathrm{Q}})\,, \tag{14}$$

where $(r_{\mathrm{Q}}, c_{\mathrm{Q}})$ and $(r_{\mathrm{K}}, c_{\mathrm{K}})$ are the row and column coordinates of the query and key patches on the $P\times P$ grid. The panel shows the joint 2D histogram of $(\Delta\text{row}, \Delta\text{col})$ with unit-width bins, revealing geometric biases such as $(0, 0)$ self-attention, fixed directional offsets, or broadly distributed global routing.

H.2 ROLE CARD EXAMPLES

We provide additional role-card examples illustrating the functional behaviors captured by nodes across different QK trees.



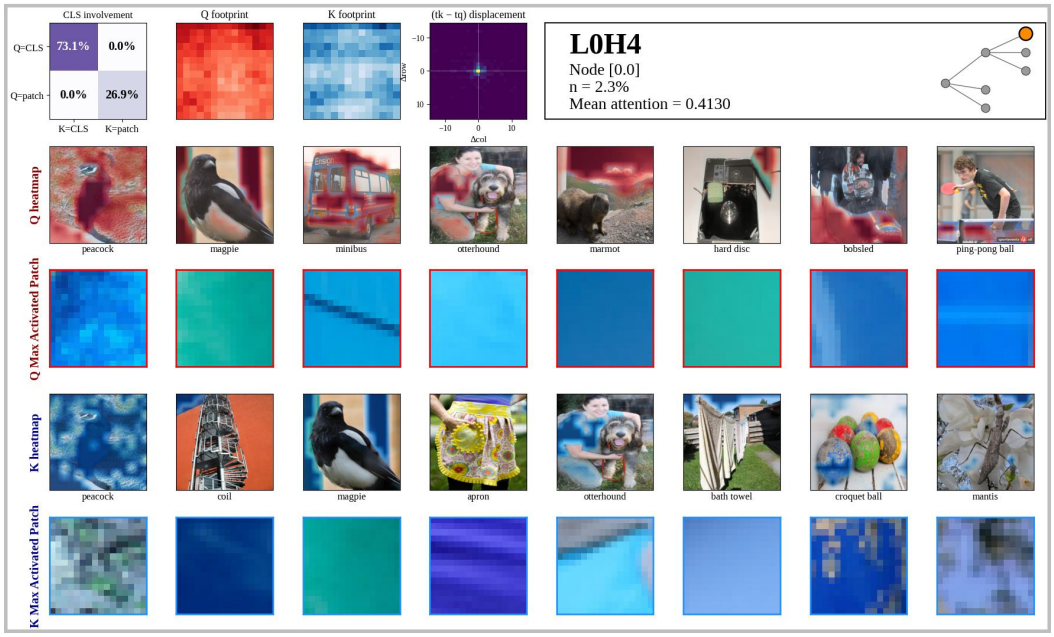


Figure 15: Example role cards of ViT-L0H4 (nodes [0] and [0.0]).

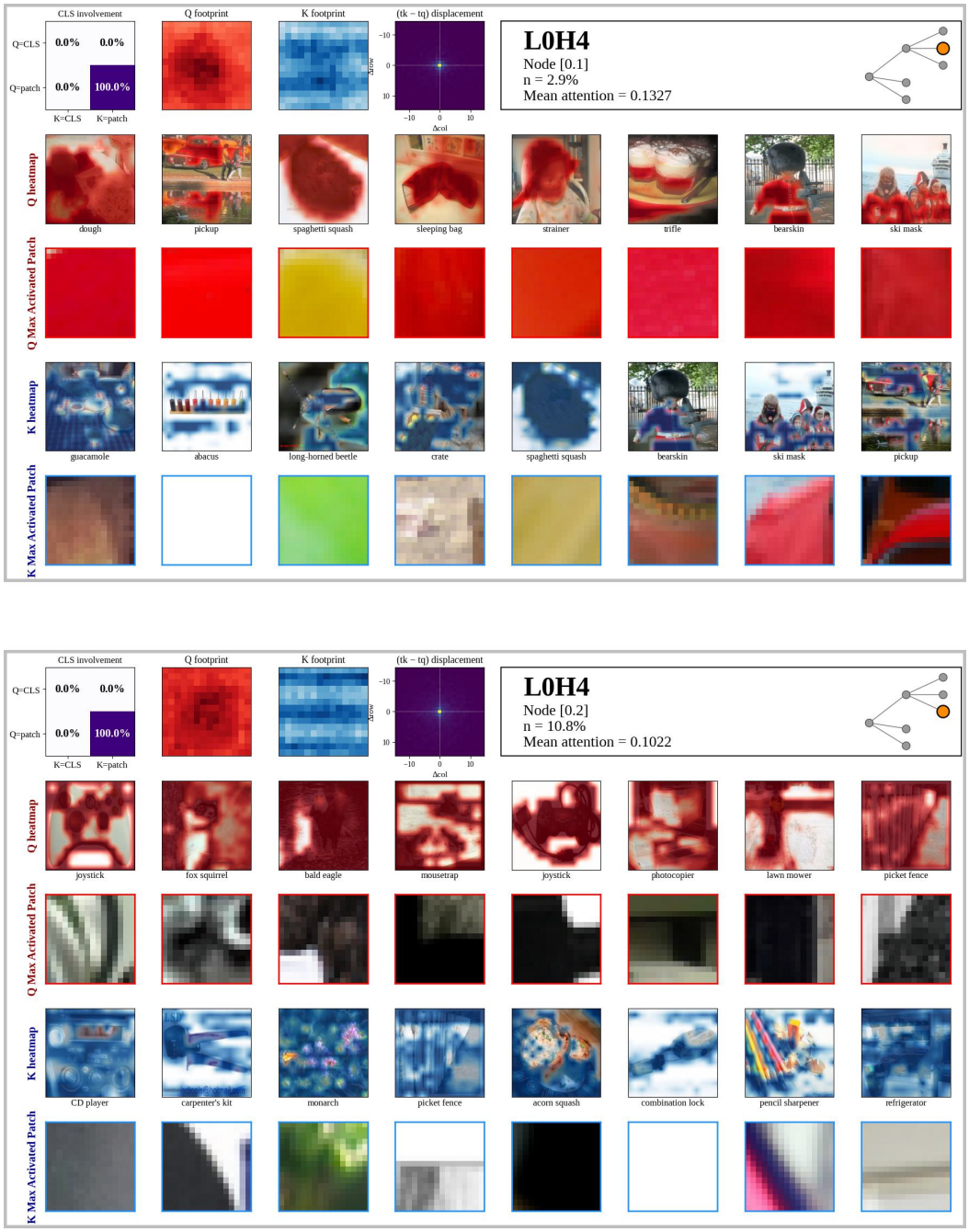


Figure 16: Example role cards of ViT-L0H4 (nodes [0.1] and [0.2]).

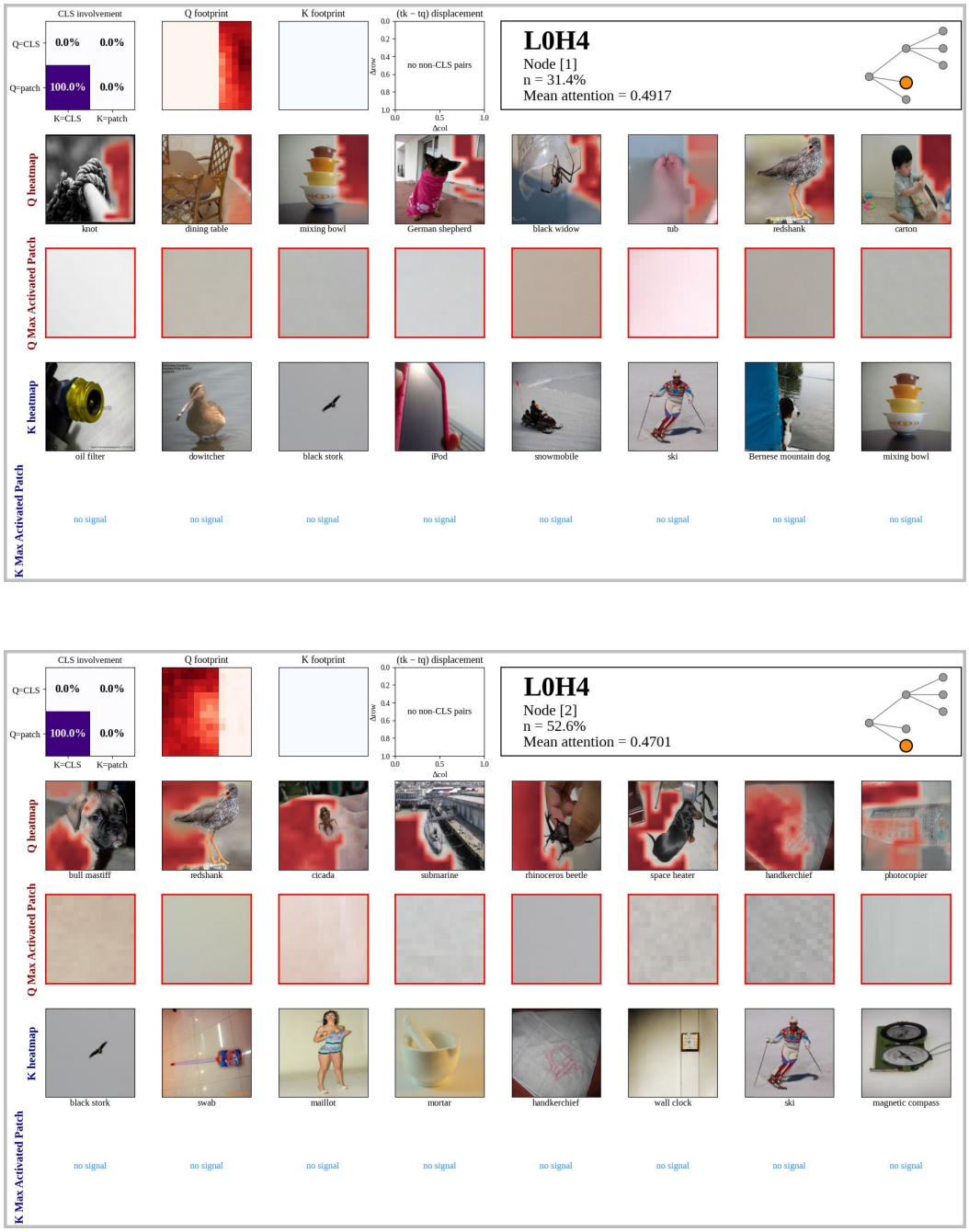


Figure 17: Example role cards of ViT-L0H4 (nodes [1] and [2]).

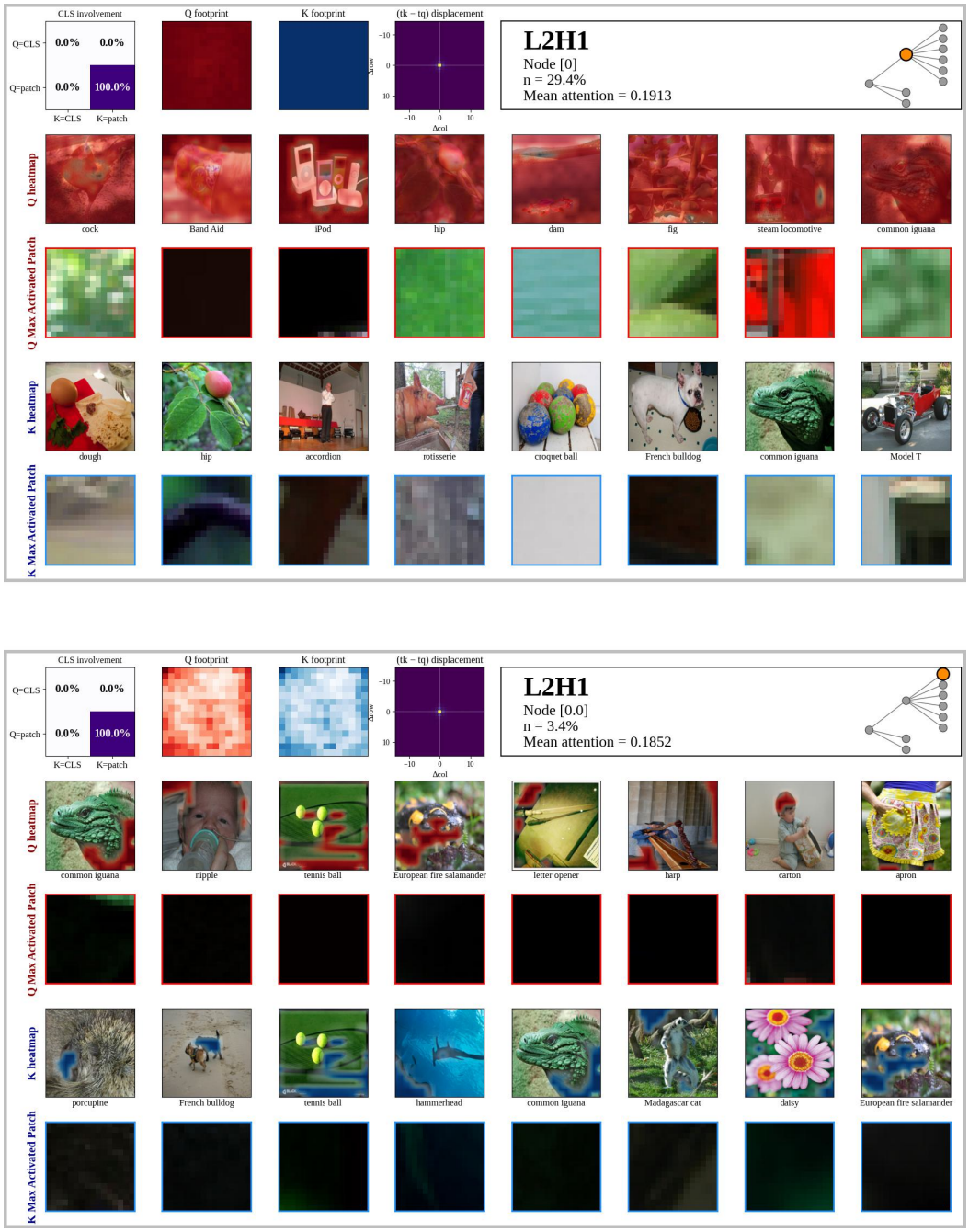


Figure 18: Example role cards of ViT-L2H1 (nodes [0] and [0.0]).

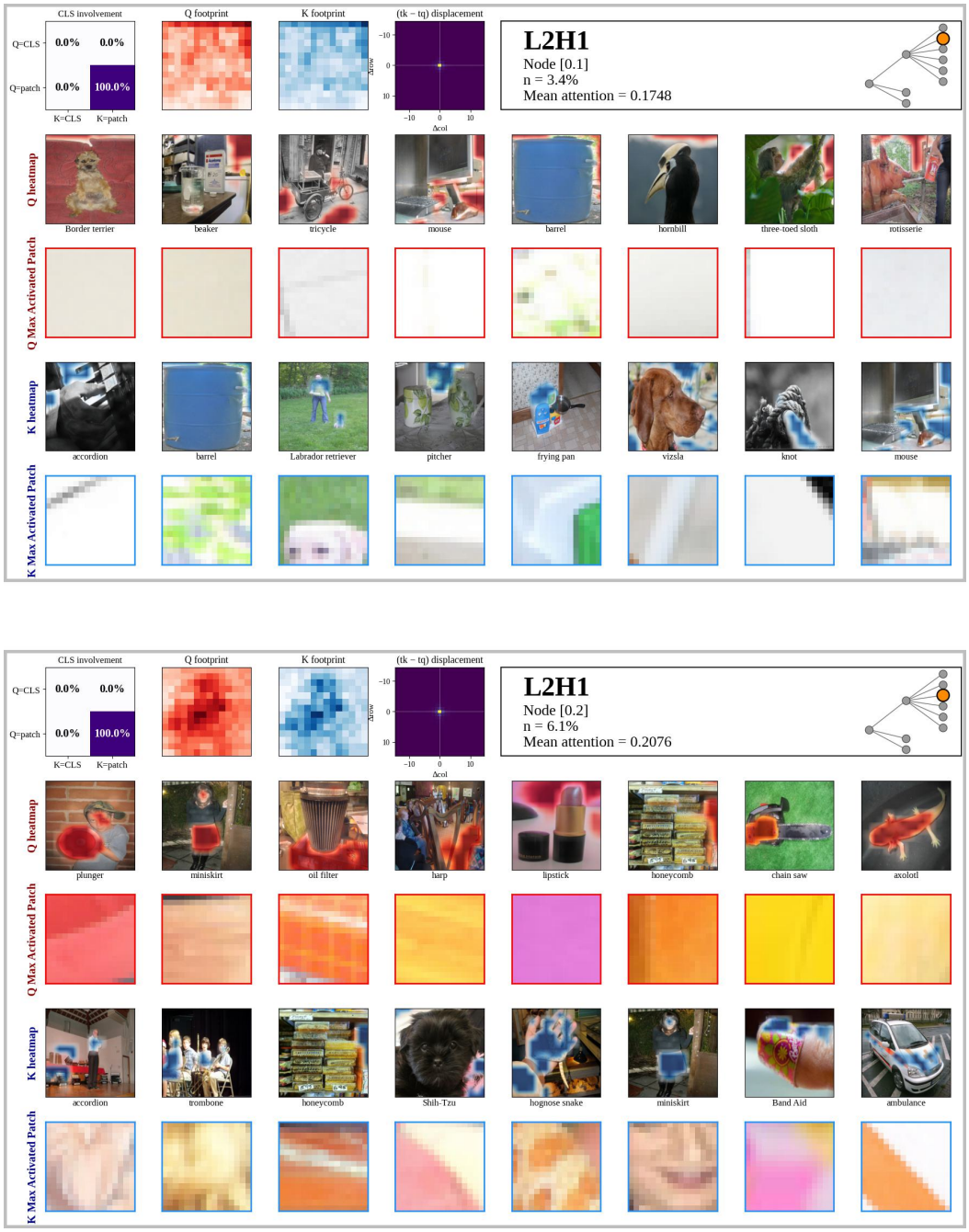


Figure 19: Example role cards of ViT-L2H1 (nodes [0.1] and [0.2]).

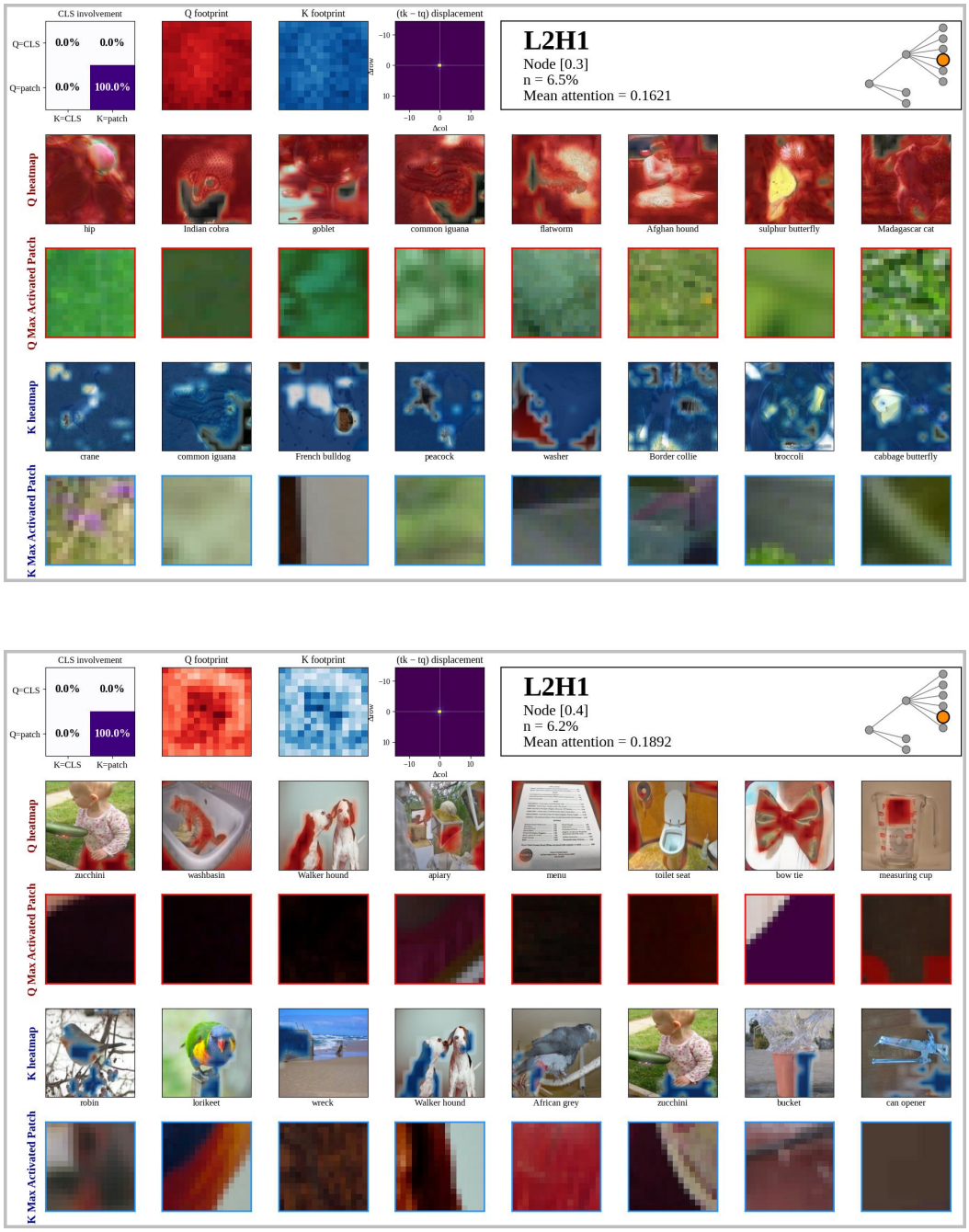


Figure 20: Example role cards of ViT-L2H1 (nodes [0.3] and [0.4]).

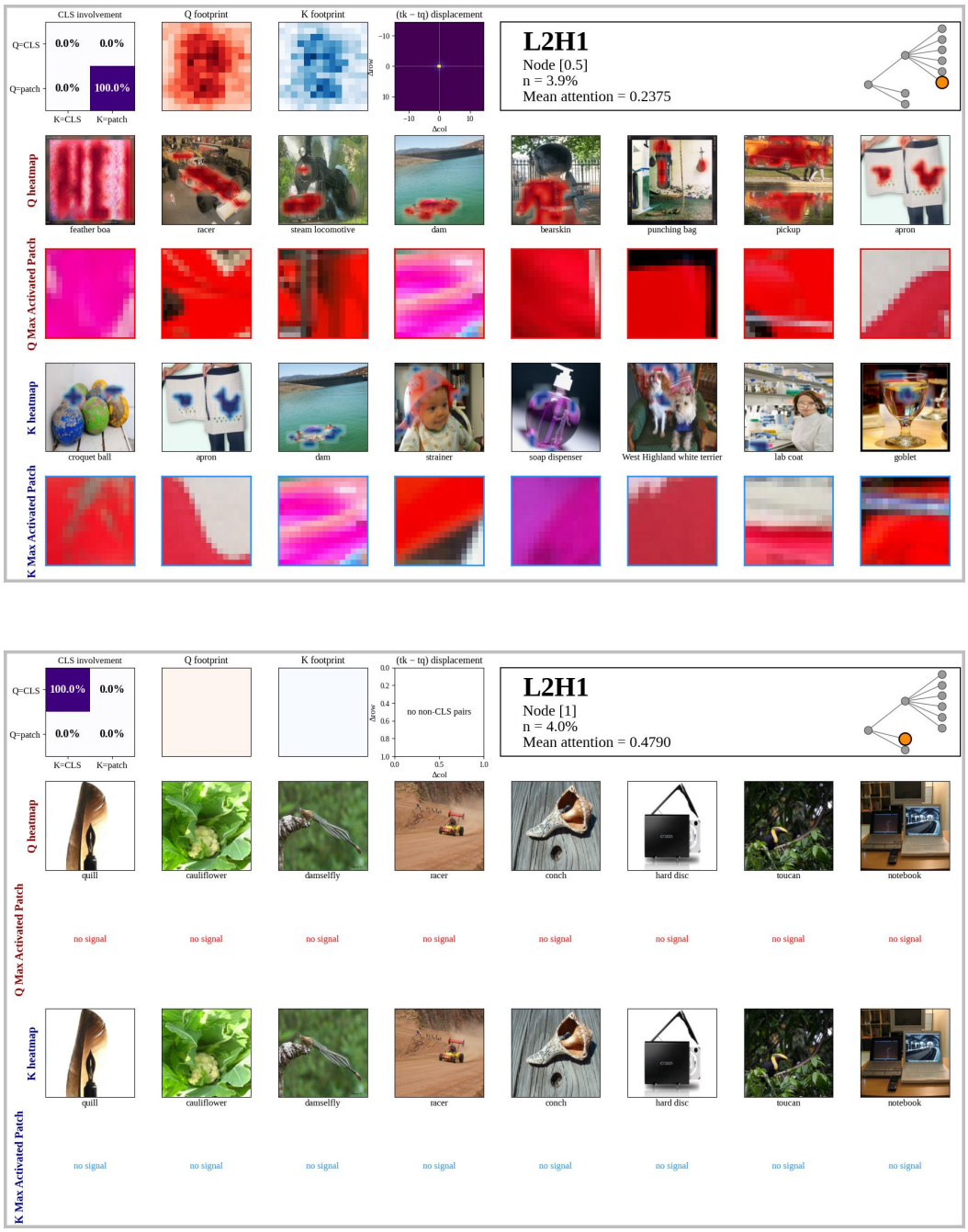


Figure 21: Example role cards of ViT-L2H1 (nodes [0.5] and [1]).



Figure 22: Example role card of ViT-L2H1 (node [2]).

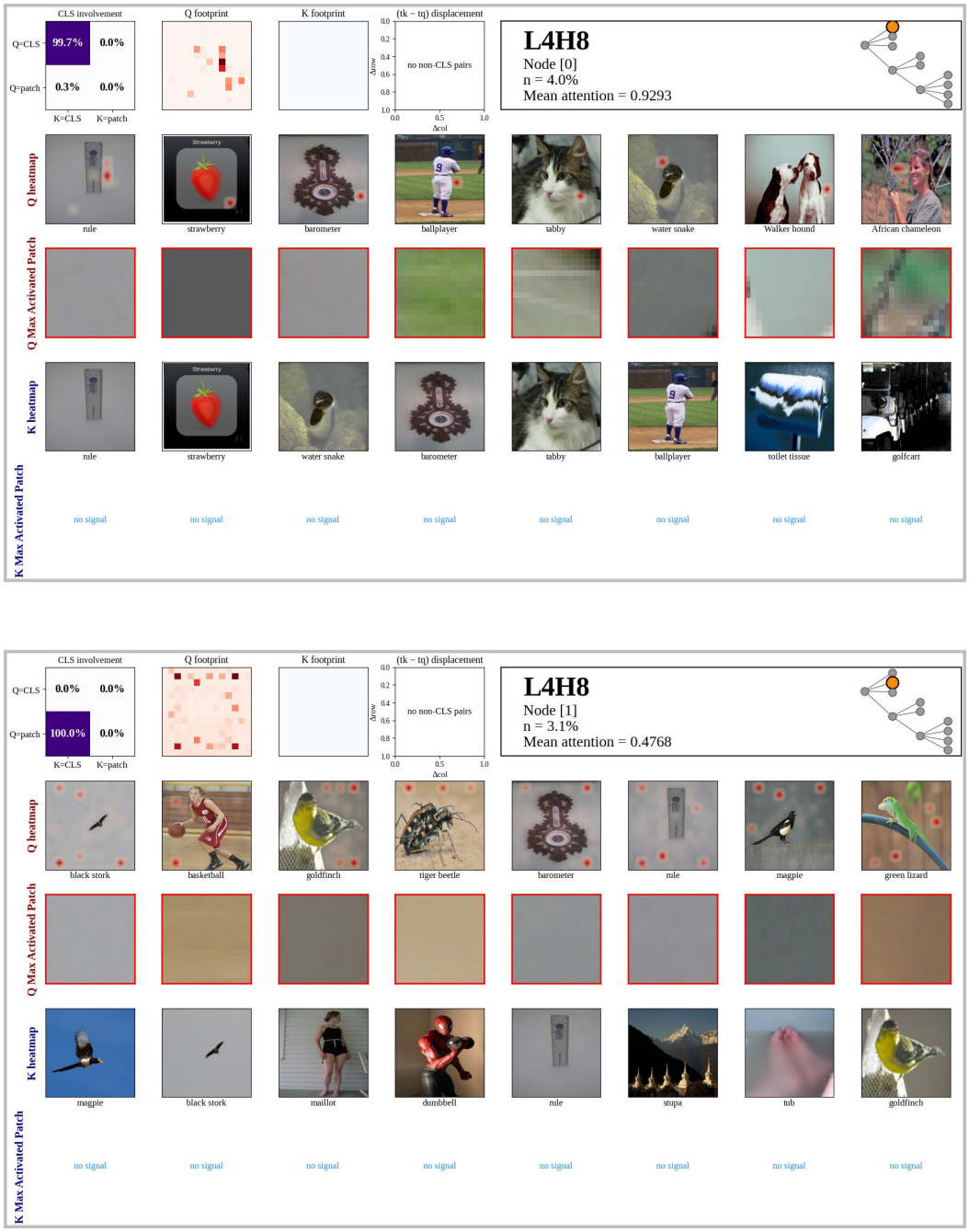


Figure 23: Example role cards of ViT-L4H8 (nodes [0] and [1]).

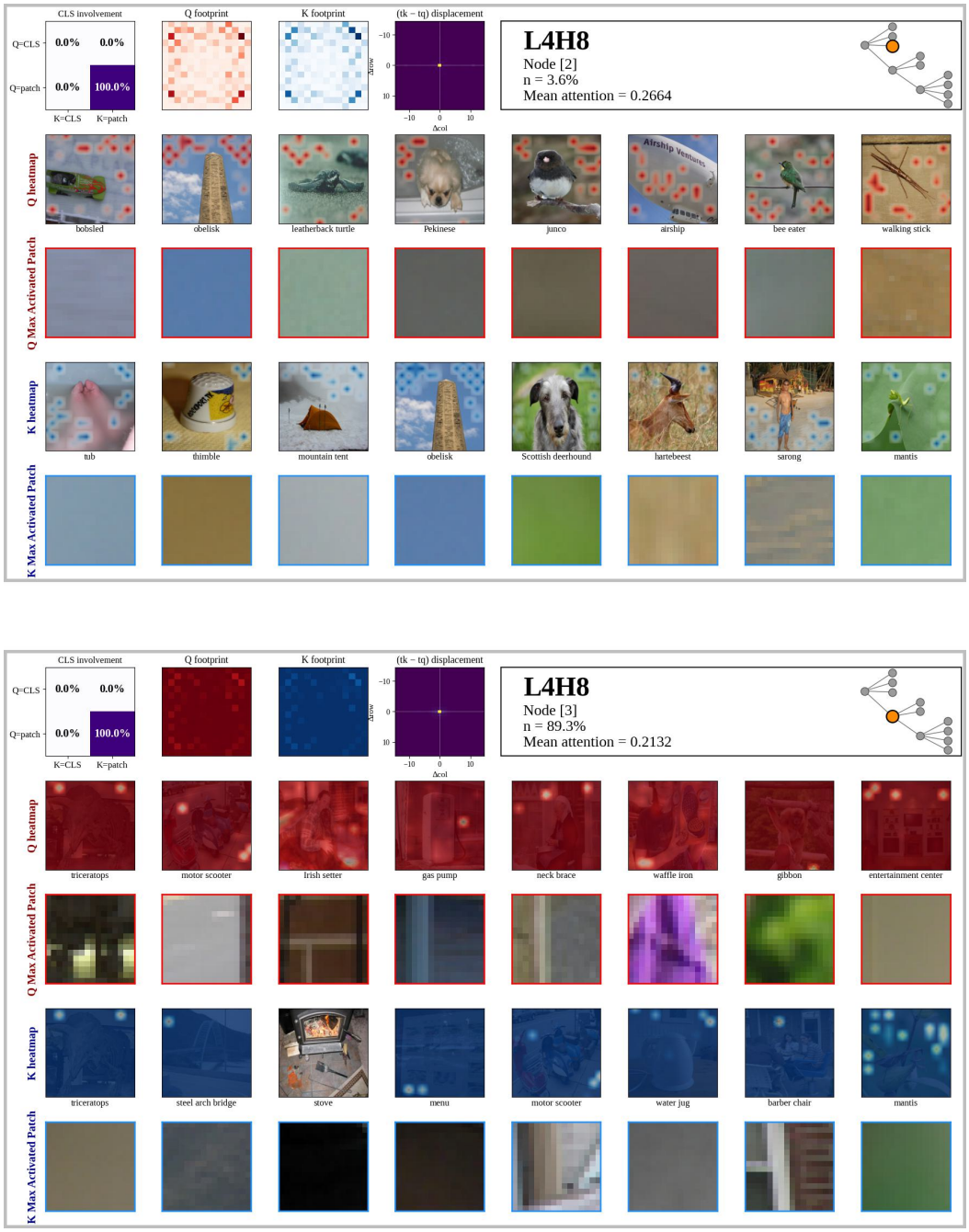
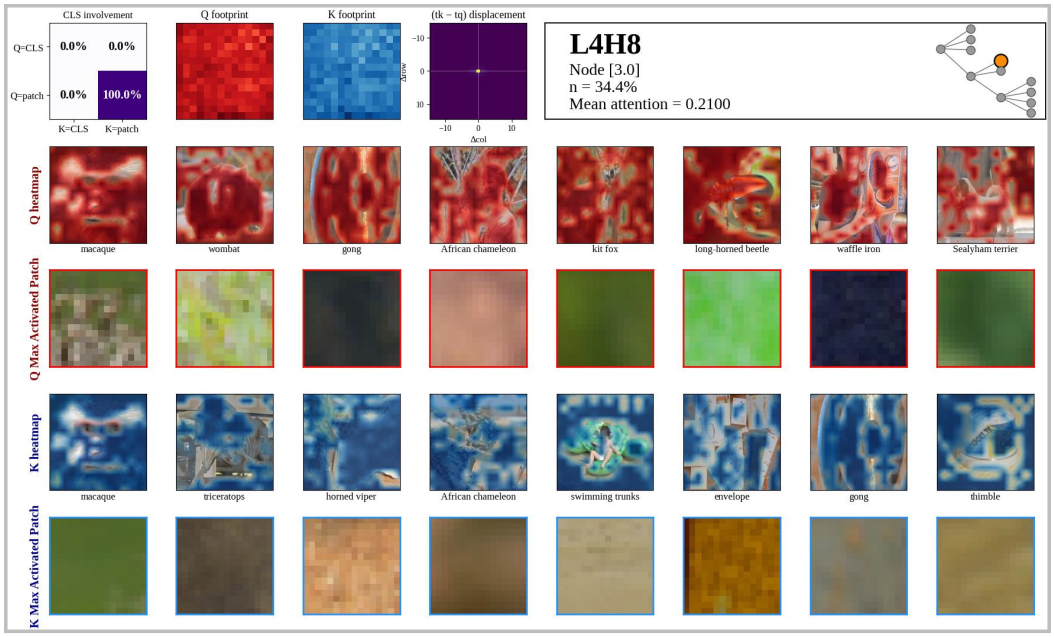


Figure 24: Example role cards of ViT-L4H8 (nodes [2] and [3]).



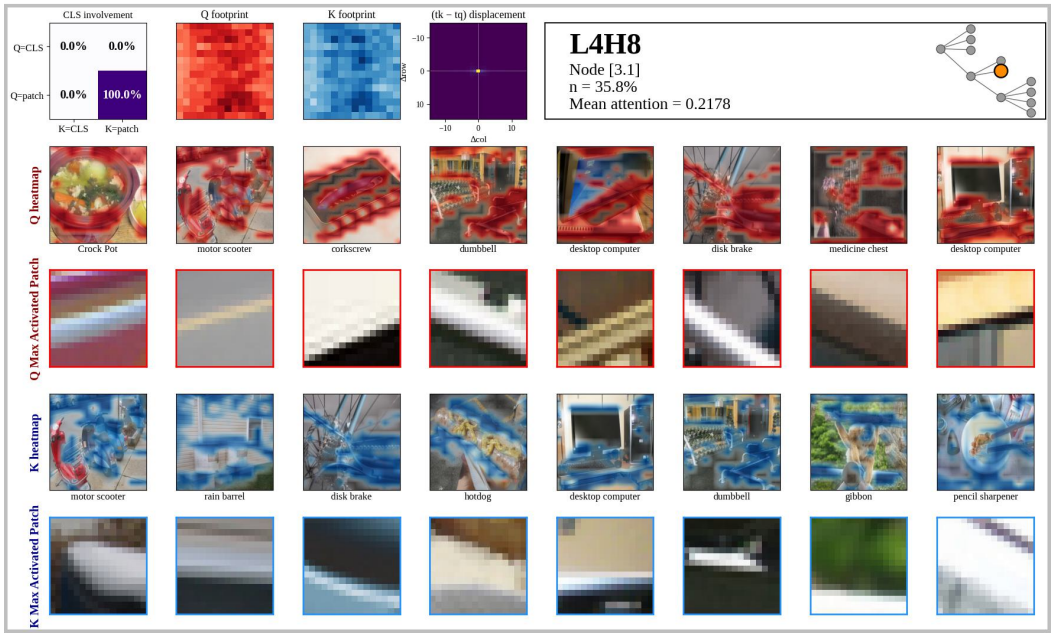


Figure 25: Example role cards of ViT-L4H8 (nodes [3.0] and [3.1]).

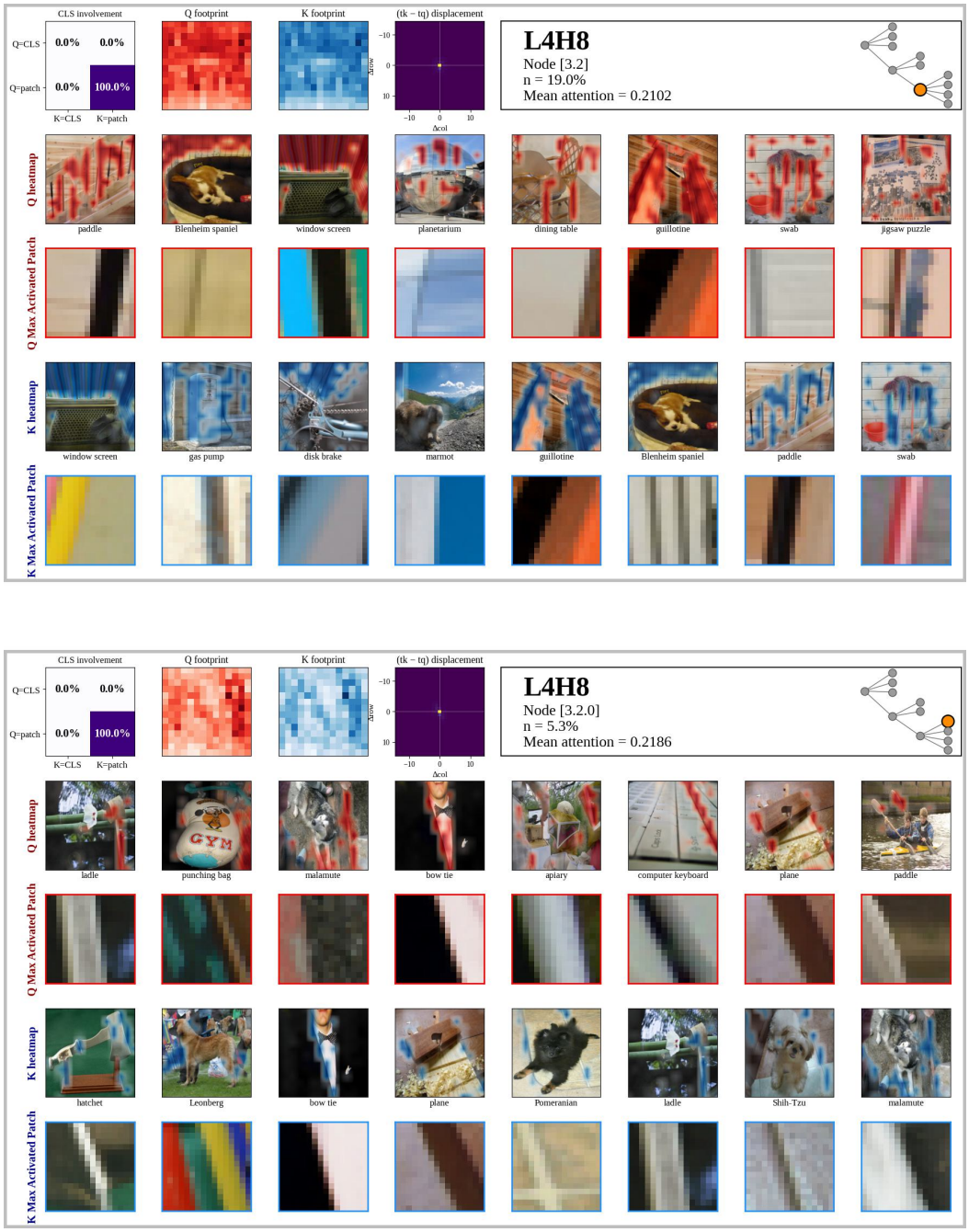


Figure 26: Example role cards of ViT-L4H8 (nodes [3.2] and [3.2.0]).

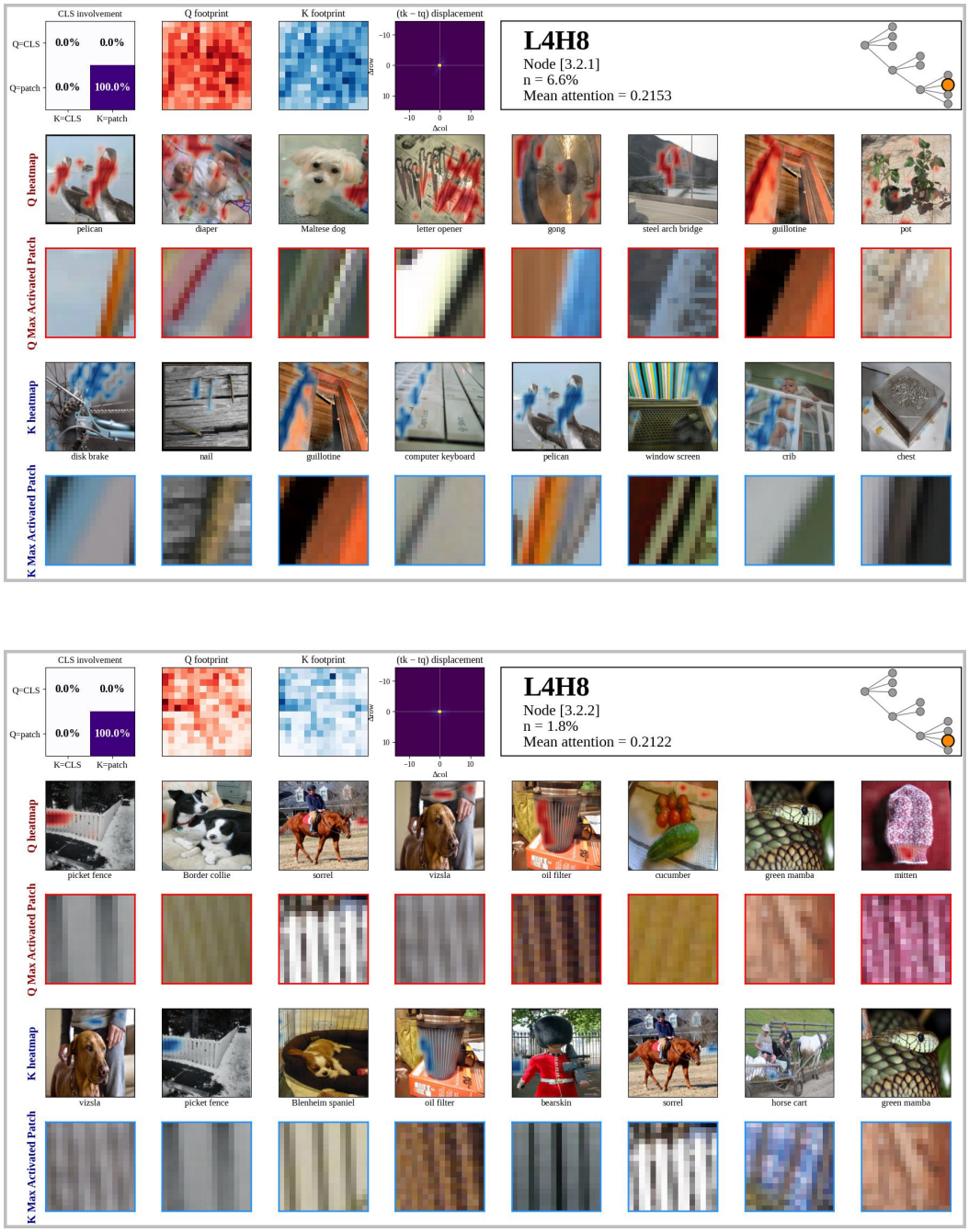


Figure 27: Example role cards of ViT-L4H8 (nodes [3.2.1] and [3.2.2]).

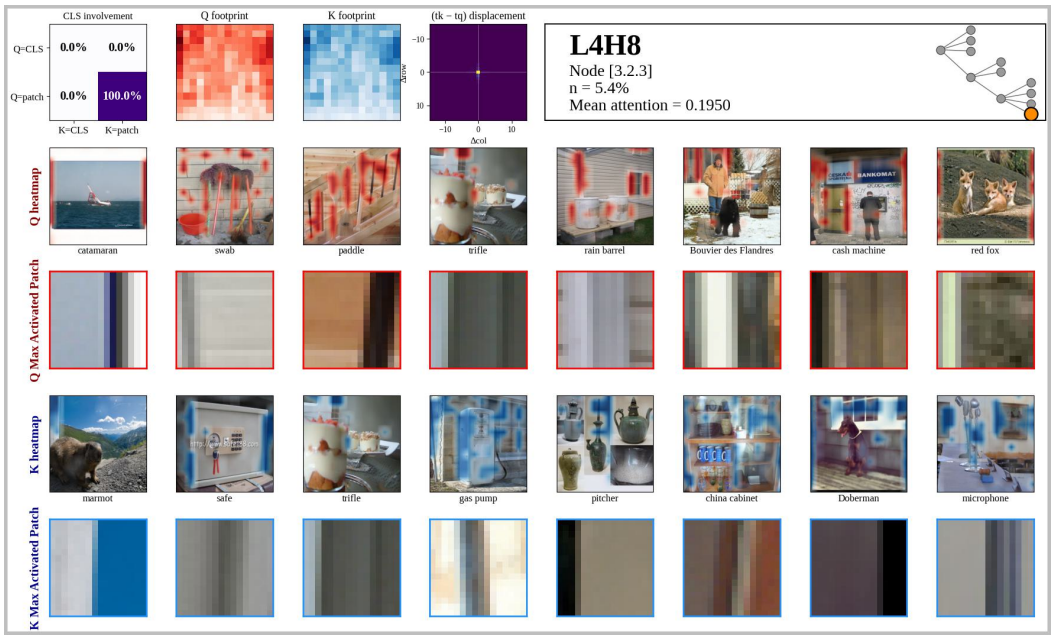


Figure 28: Example role card of ViT-L4H8 (node [3.2.3]).

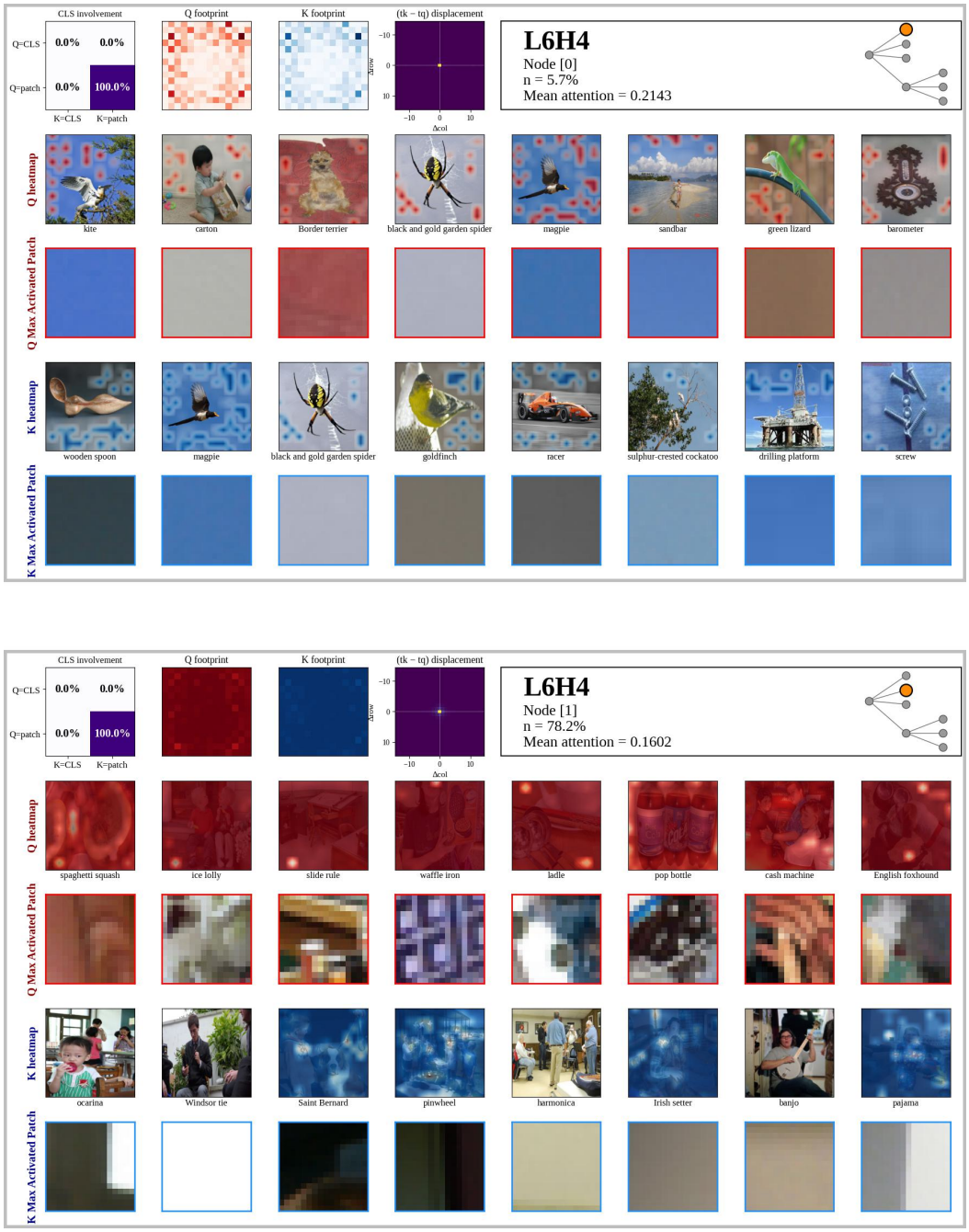


Figure 29: Example role cards of ViT-L6H4 (nodes [0] and [1]).

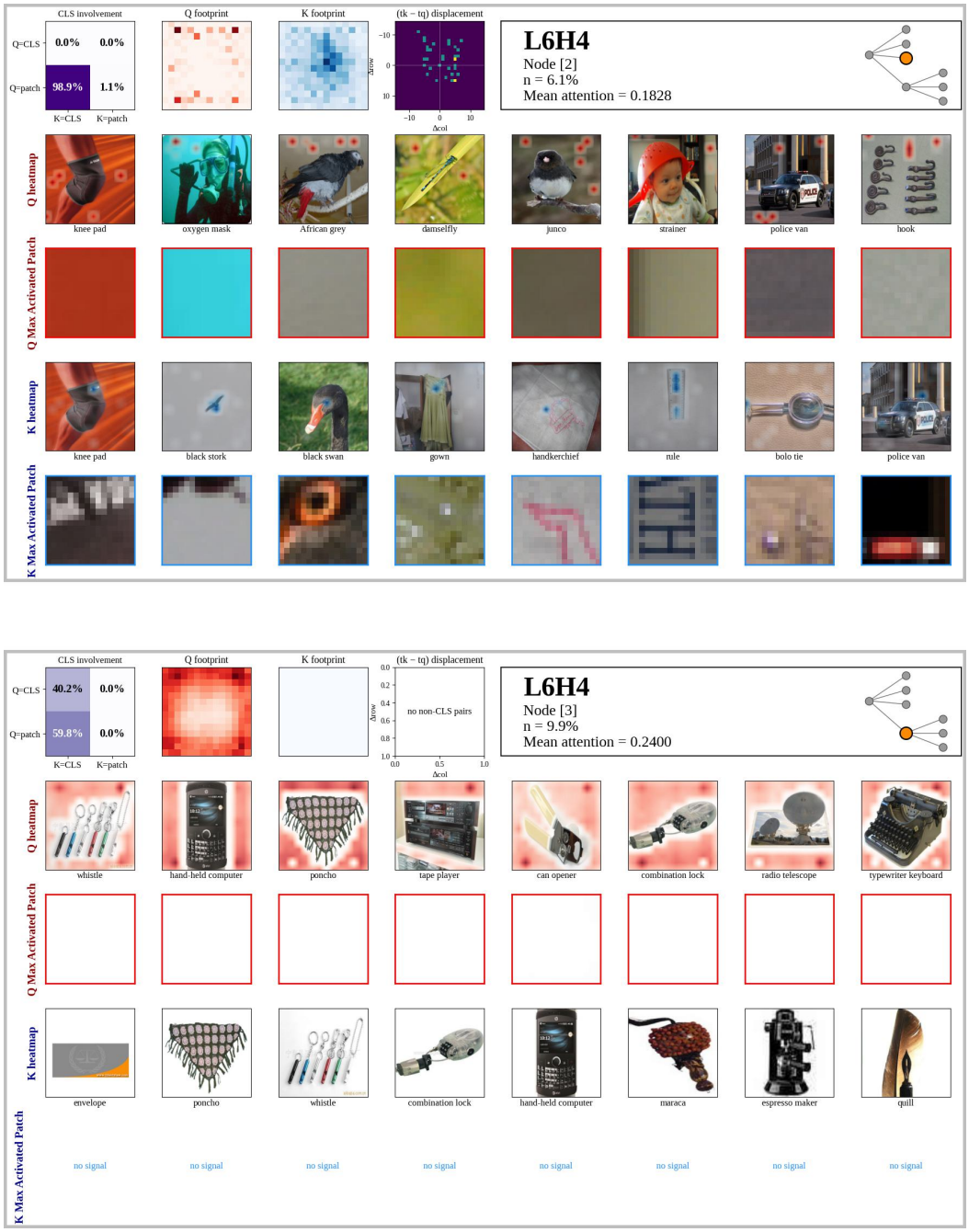


Figure 30: Example role cards of ViT-L6H4 (nodes [2] and [3]).

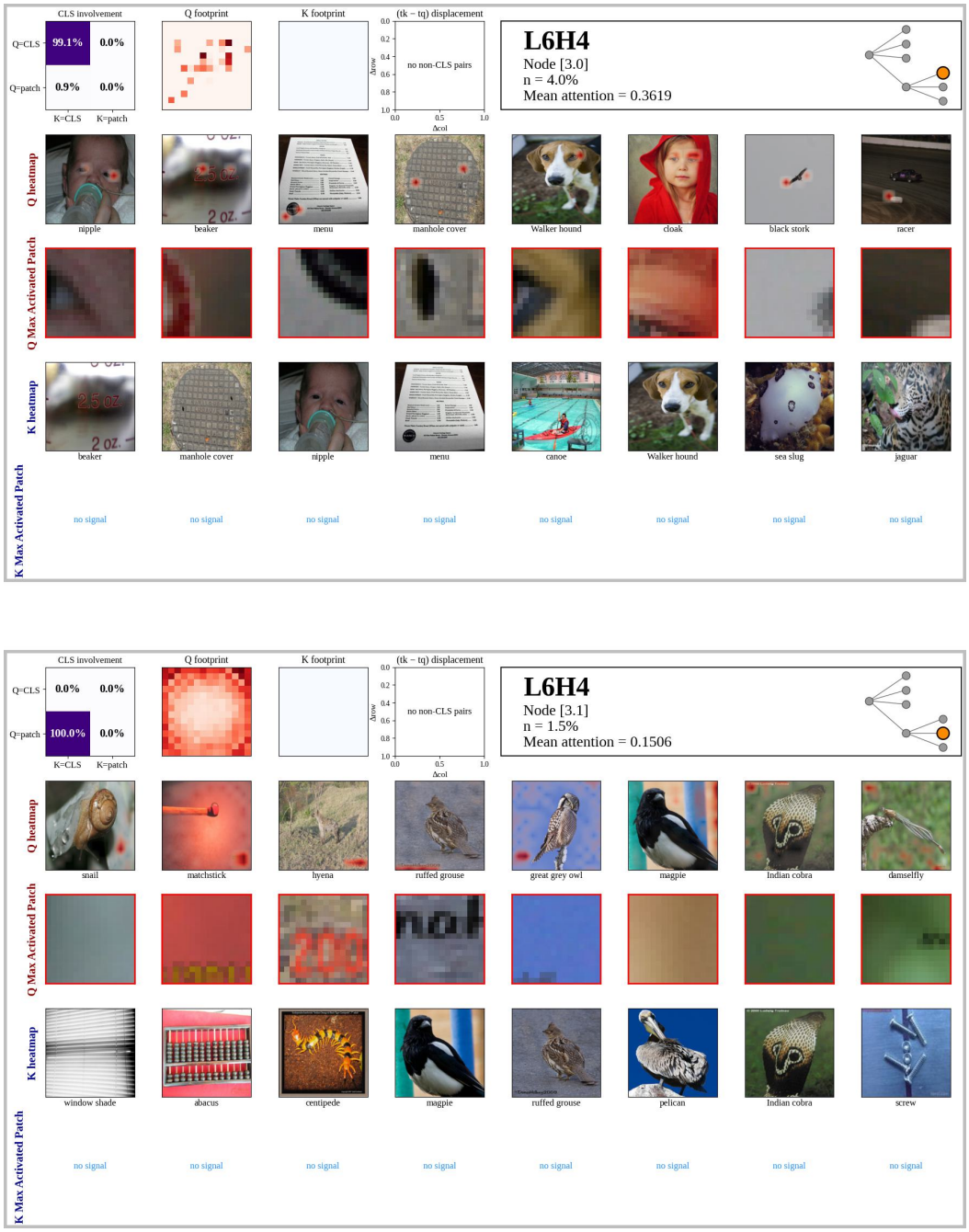


Figure 31: Example role cards of ViT-L6H4 (nodes [3.0] and [3.1]).

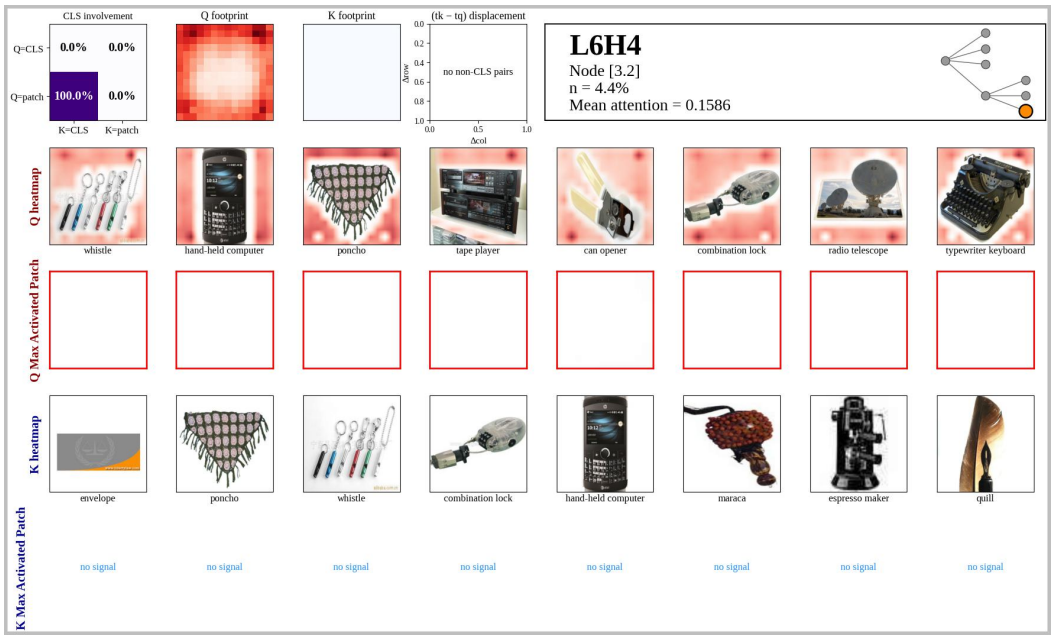


Figure 32: Example role card of ViT-L6H4 (node [3.2]).

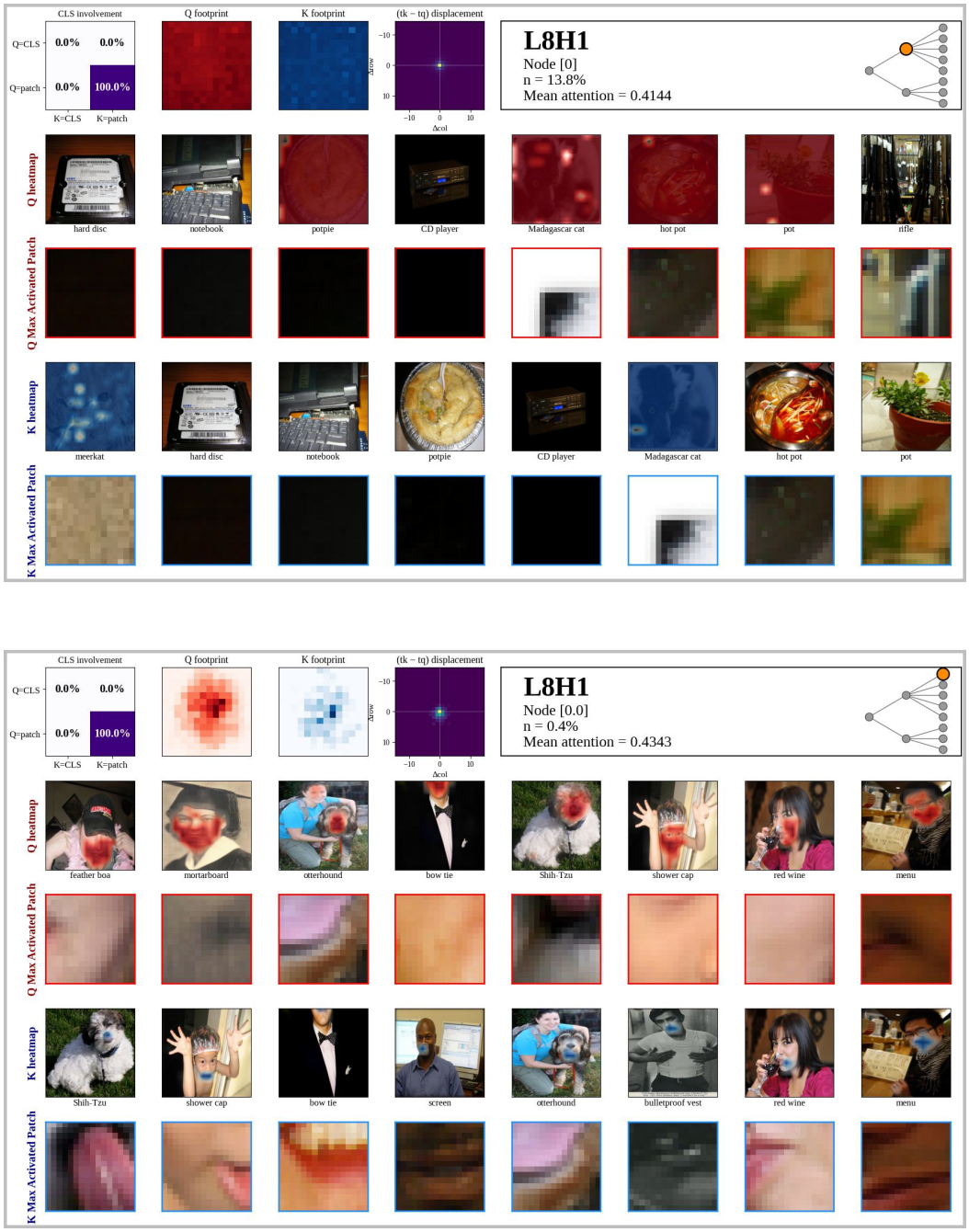


Figure 33: Example role cards of ViT-L8H1 (nodes [0] and [0.0]).



Figure 34: Example role cards of ViT-L8H1 (nodes [0.1] and [0.2]).

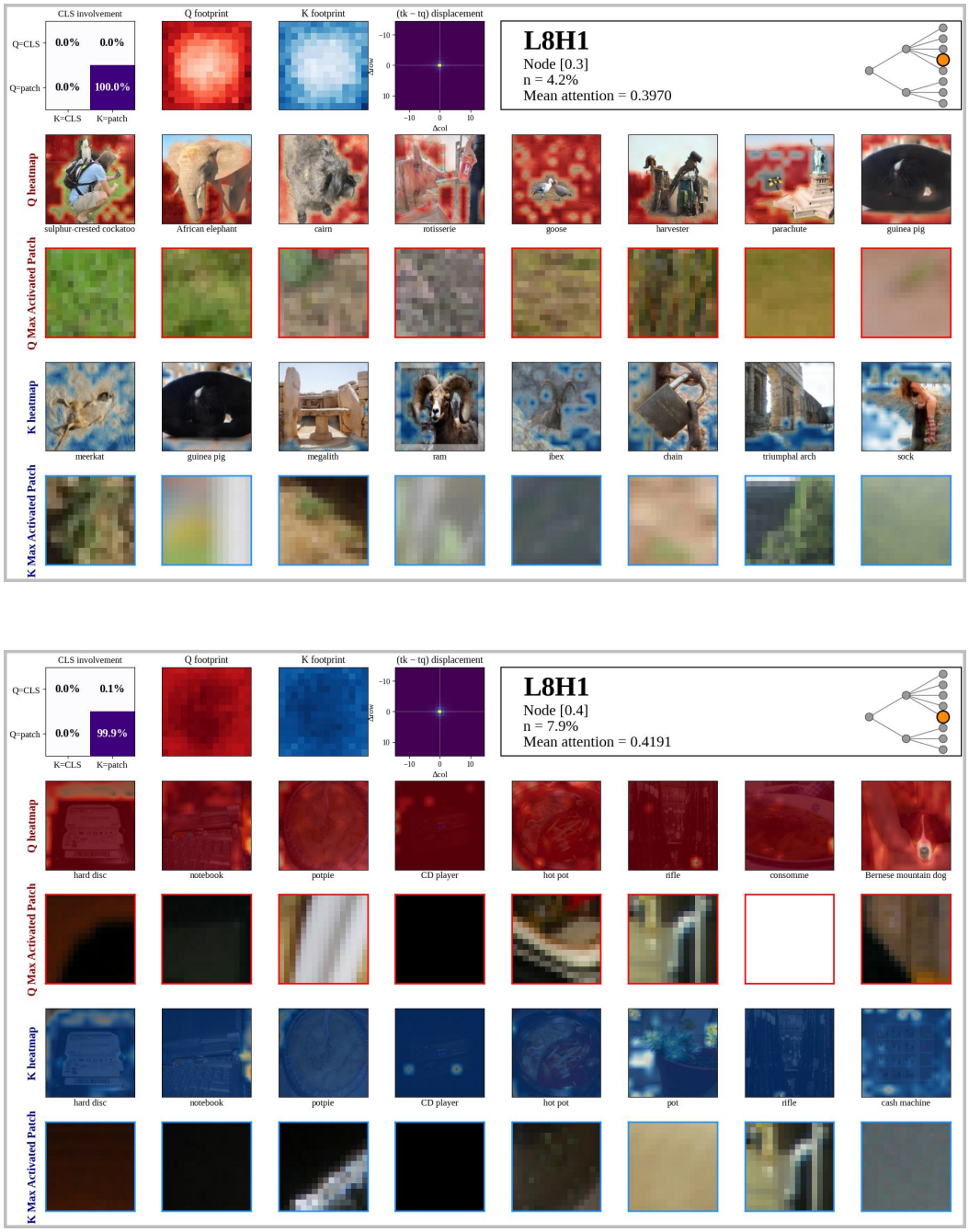


Figure 35: Example role cards of ViT-L8H1 (nodes [0.3] and [0.4]).

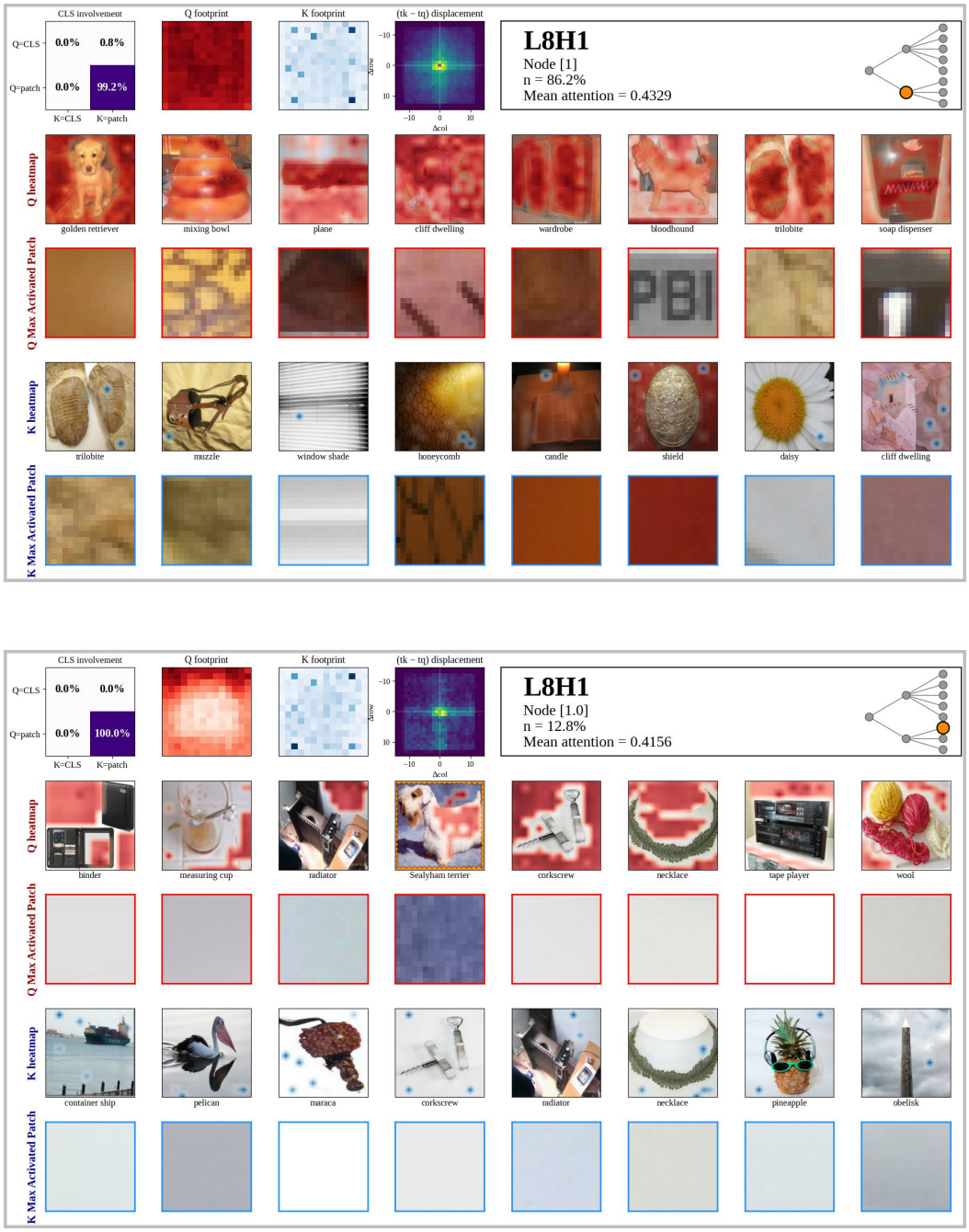


Figure 36: Example role cards of ViT-L8H1 (nodes [1] and [1.0]).

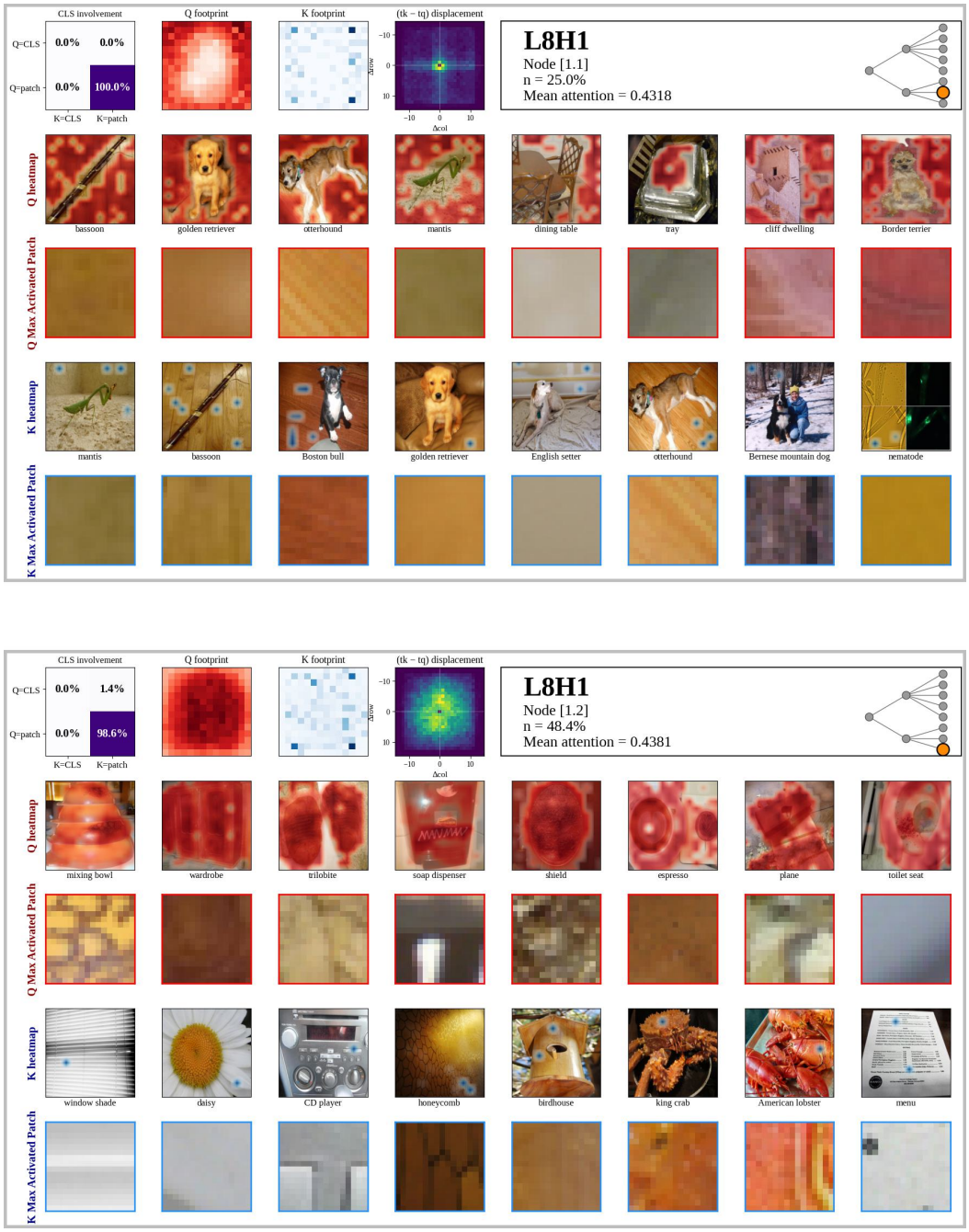
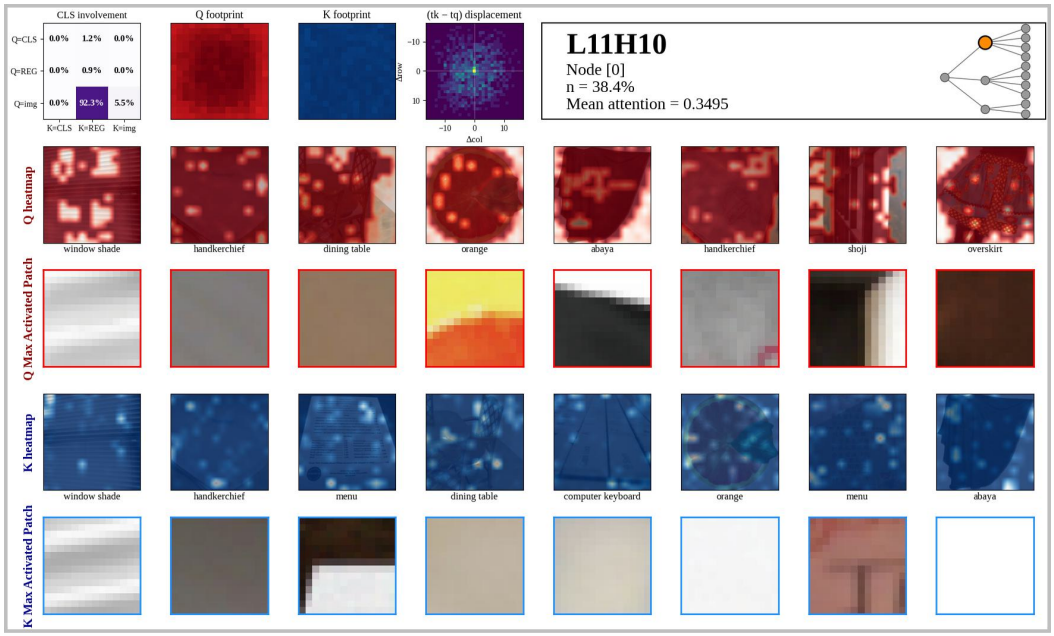


Figure 37: Example role cards of ViT-L8H1 (nodes [1.1] and [1.2]).



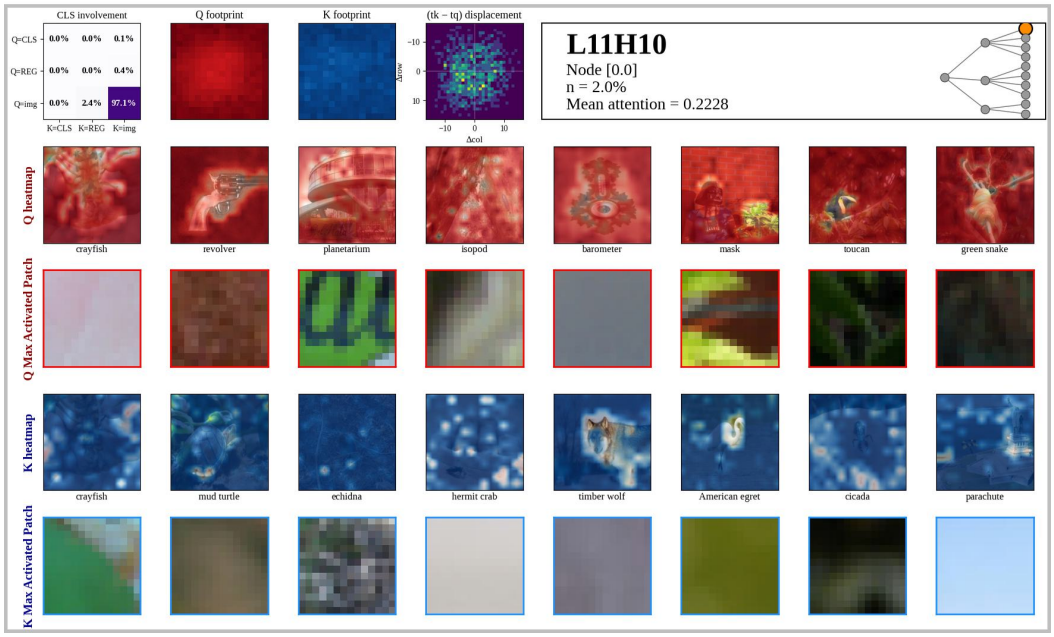
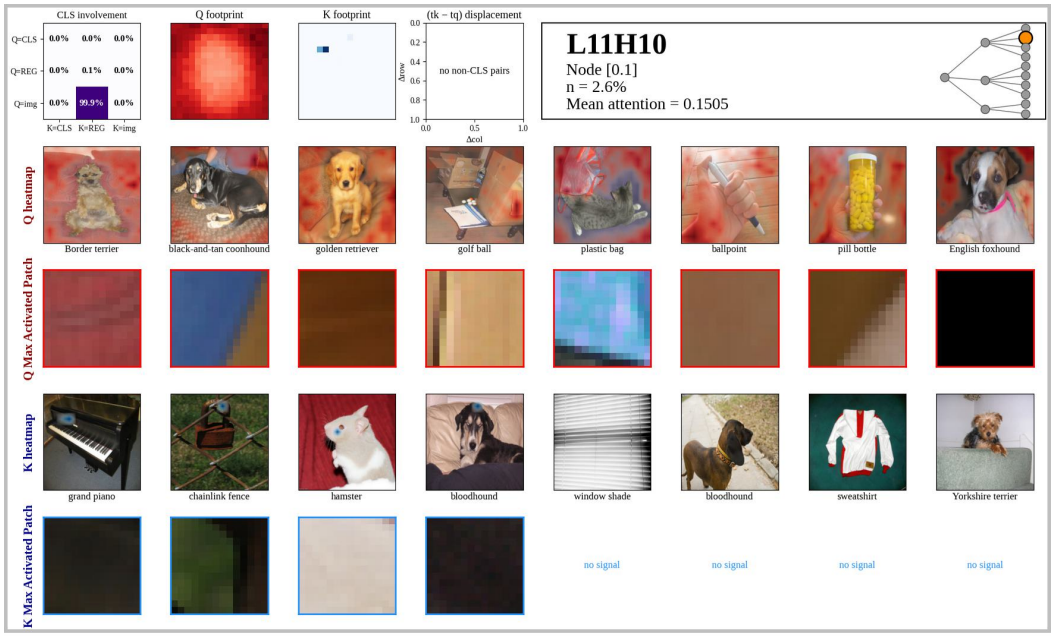


Figure 38: Example role cards of DINOv2-L11H10 (nodes [0] and [0.0]).



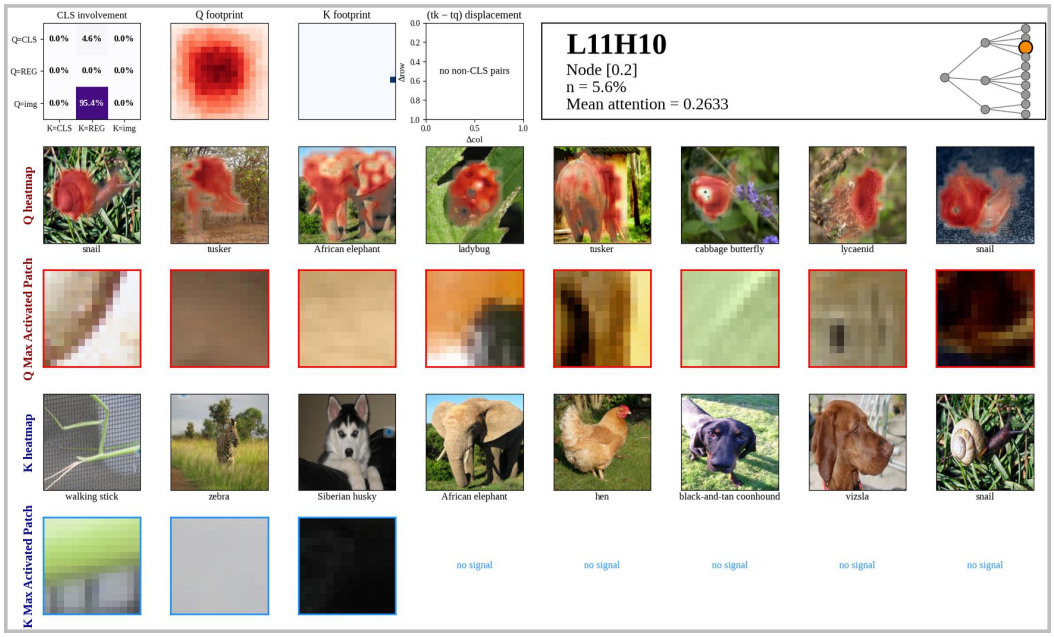


Figure 39: Example role cards of DINOv2-L11H10 (nodes [0.1] and [0.2]).

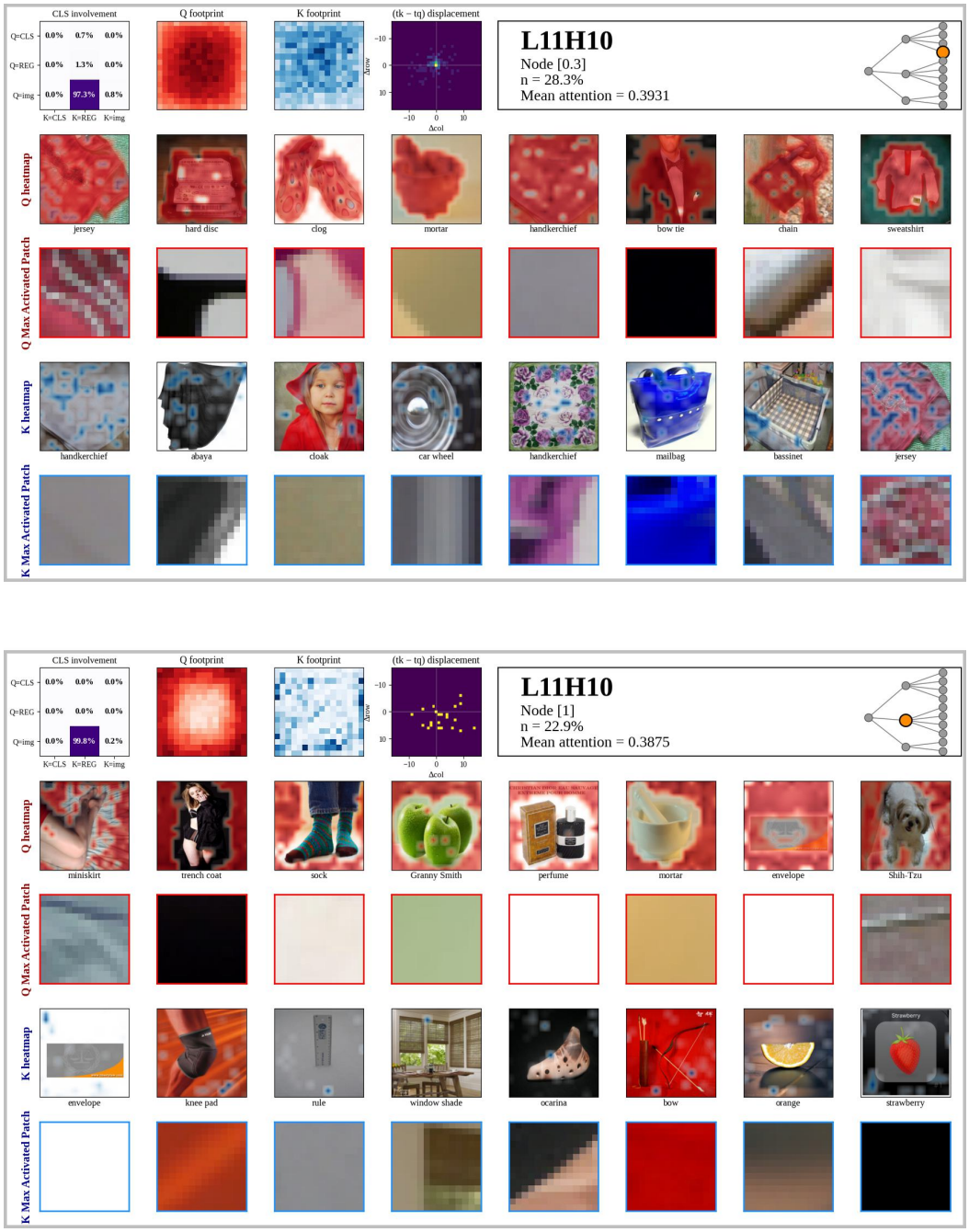
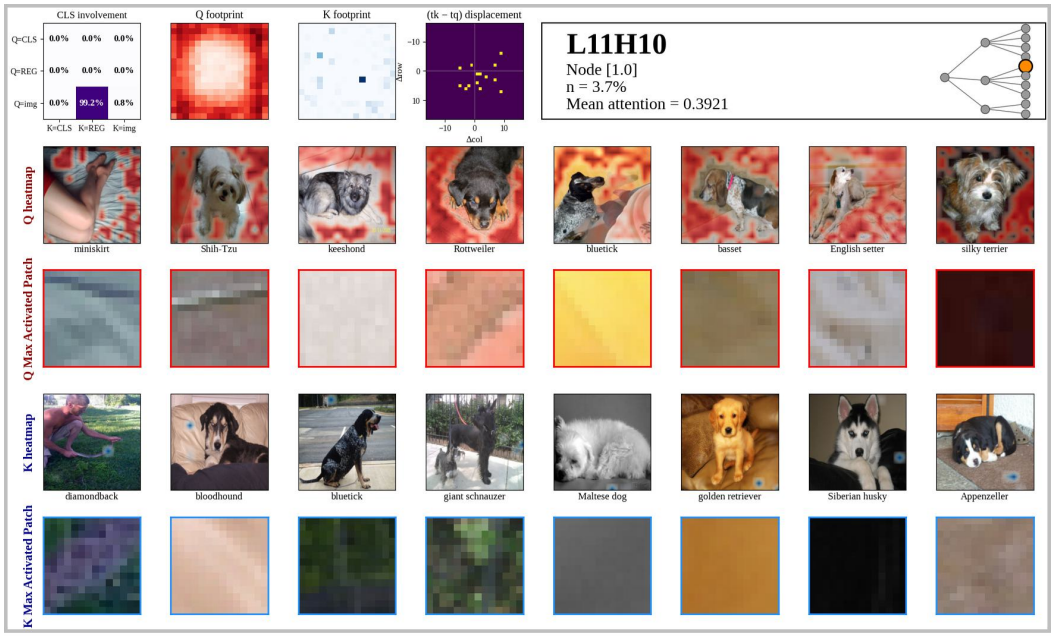


Figure 40: Example role cards of DINOv2-L11H10 (nodes [0.3] and [1]).



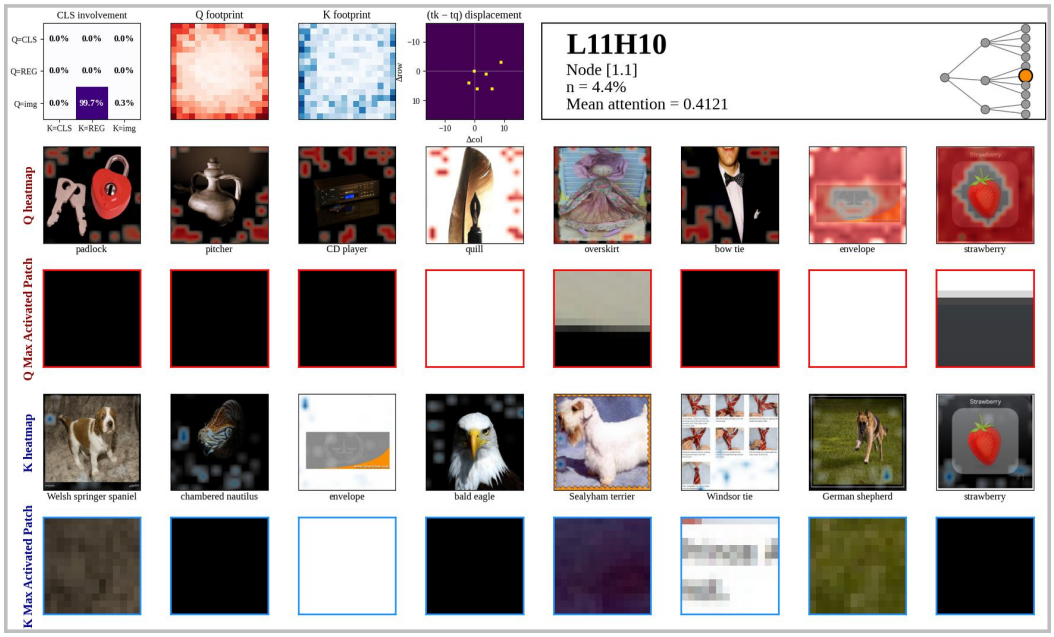


Figure 41: Example role cards of DINOv2-L11H10 (nodes [1.0] and [1.1]).

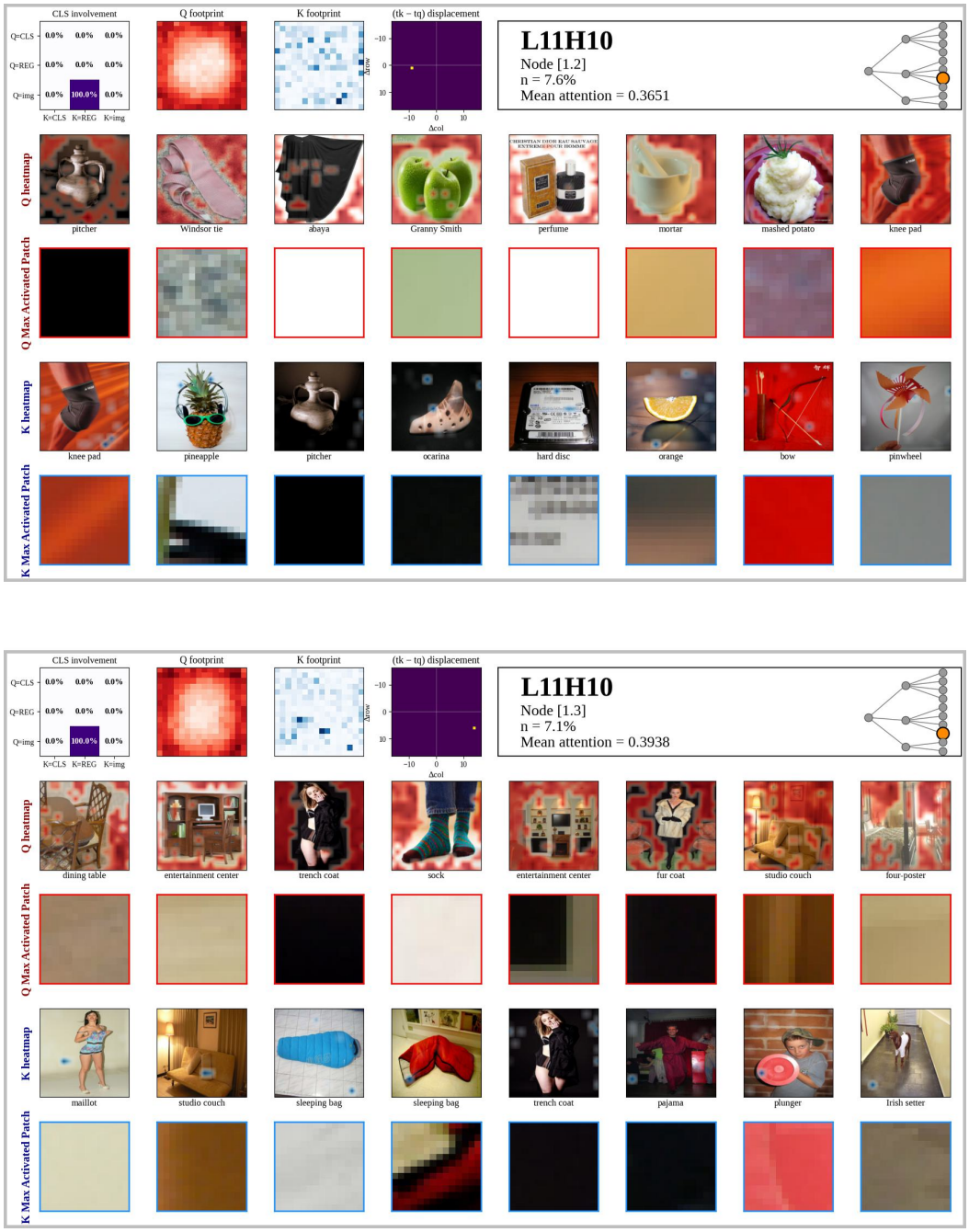
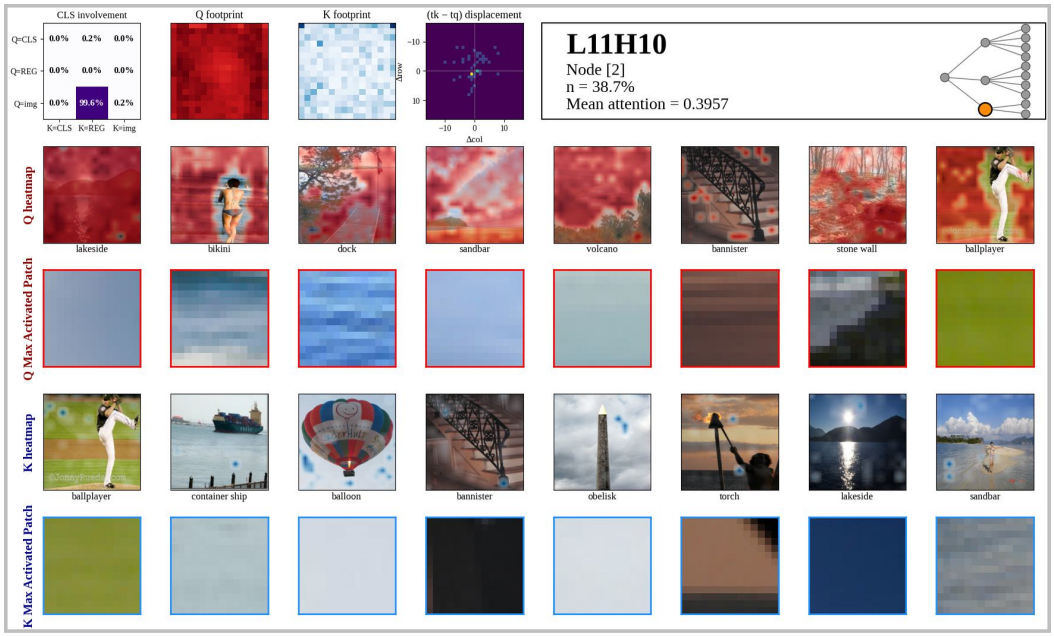


Figure 42: Example role cards of DINOv2-L11H10 (nodes [1.2] and [1.3]).



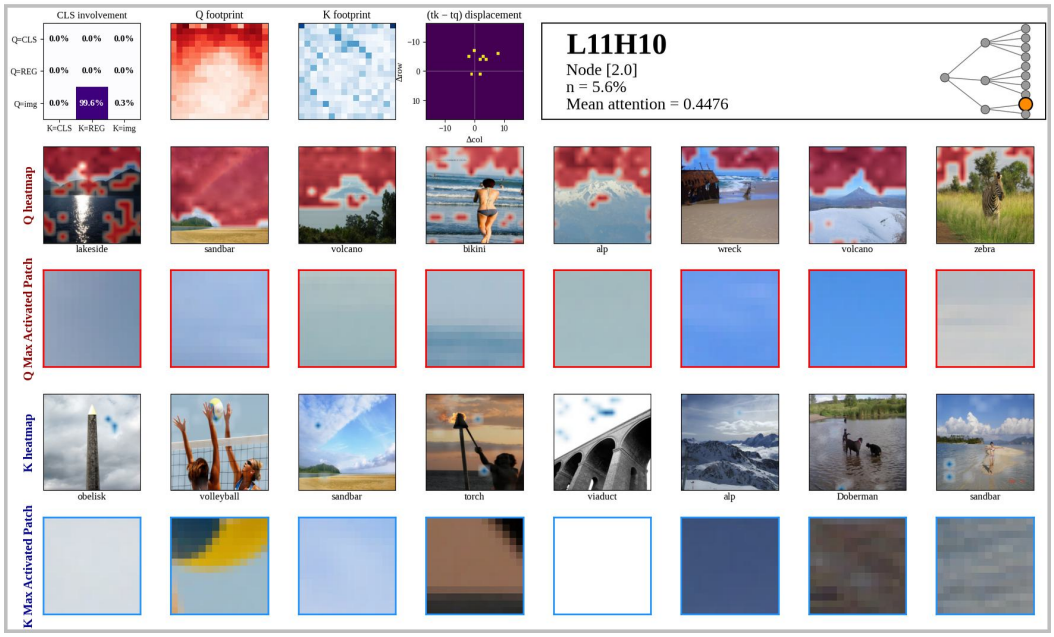
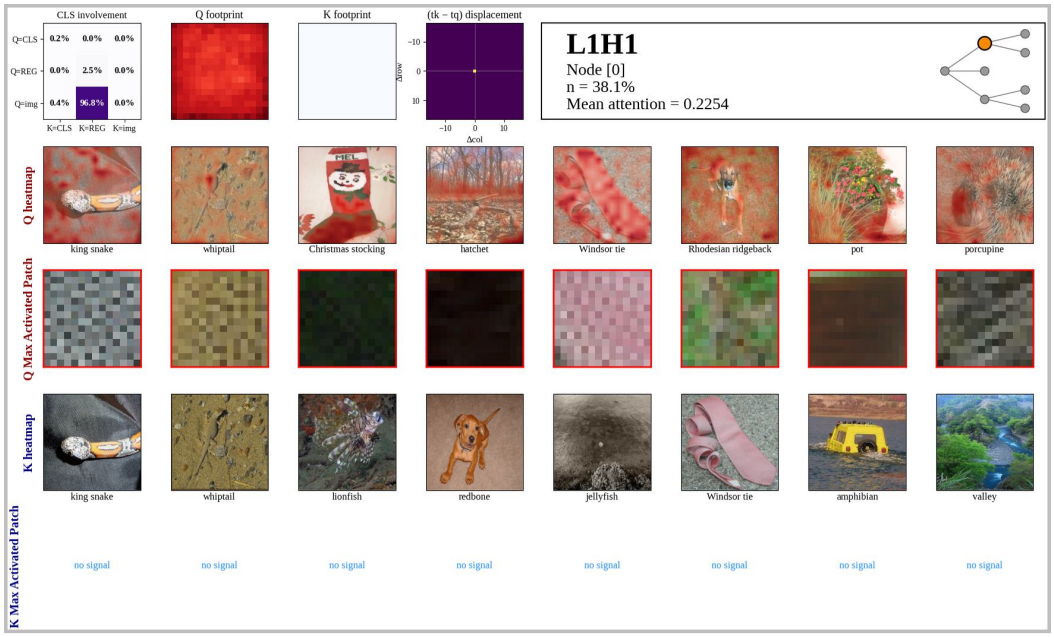


Figure 43: Example role cards of DINOv2-L11H10 (nodes [2] and [2.0]).



Figure 44: Example role card of DINOv2-L11H10 (node [2.1]).



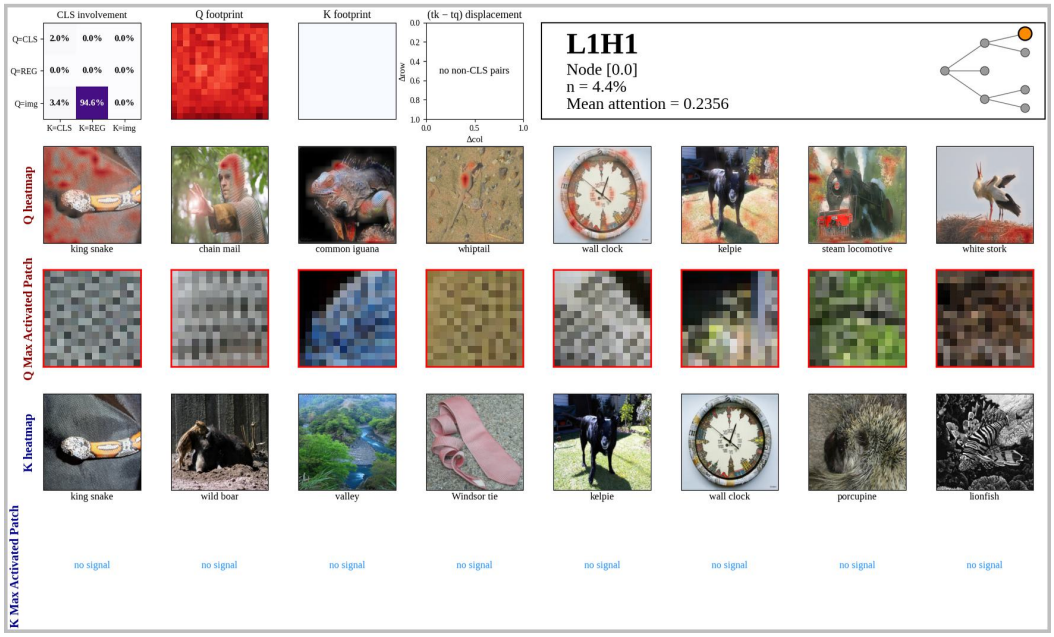


Figure 45: Example role cards of DINOv2-L1H1 (nodes [0] and [0.0]).

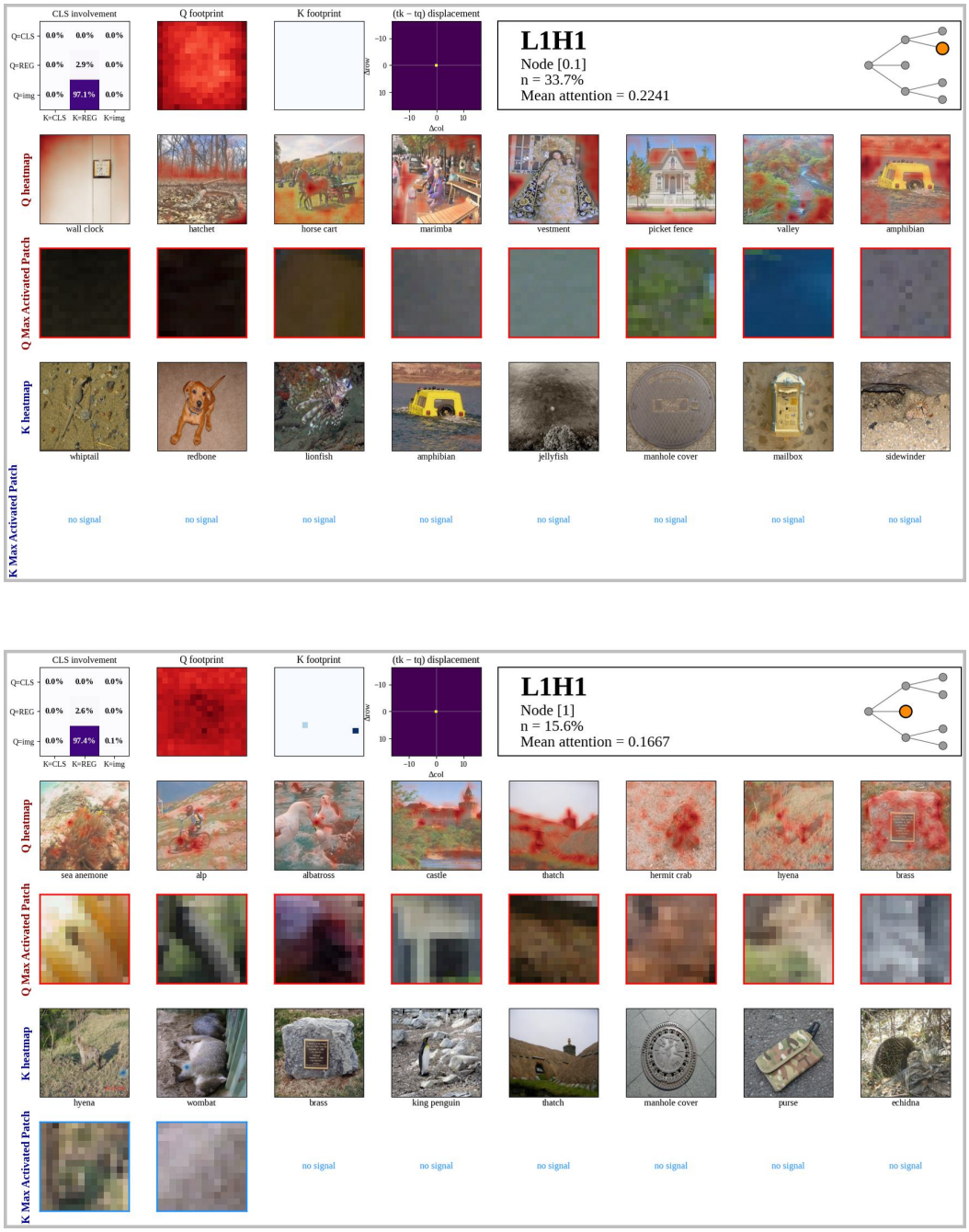
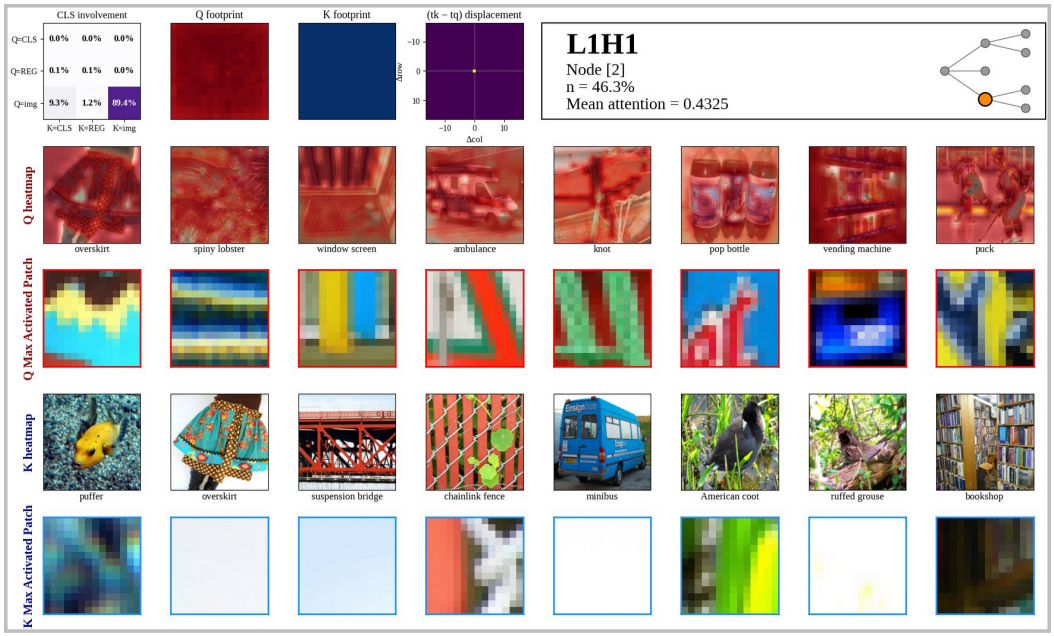


Figure 46: Example role cards of DINOv2-L1H1 (nodes [0.1] and [1]).



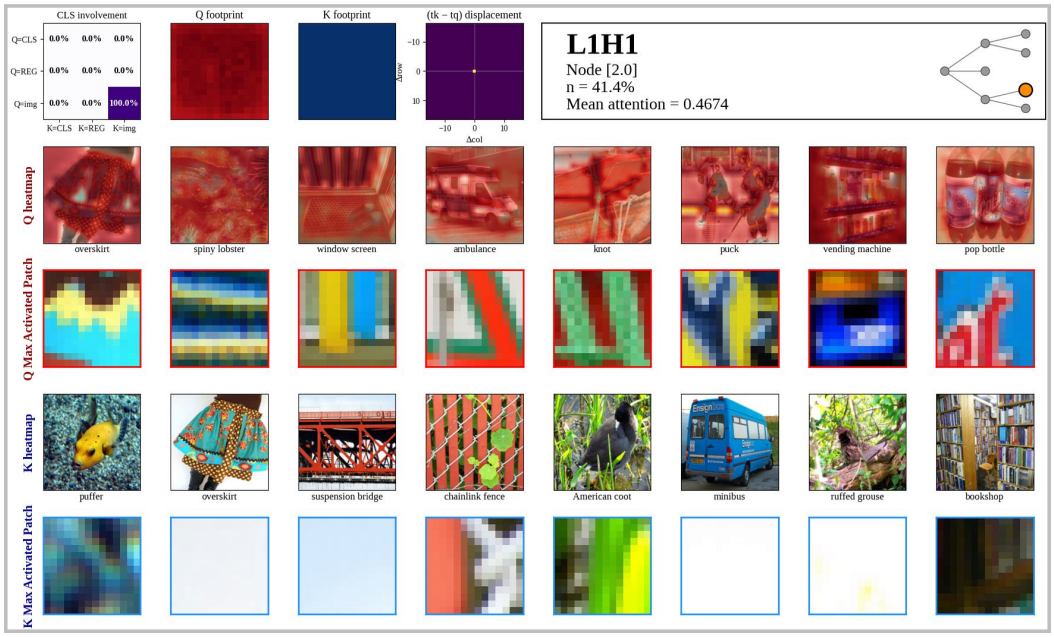


Figure 47: Example role cards of DINOv2-L1H1 (nodes [2] and [2.0]).

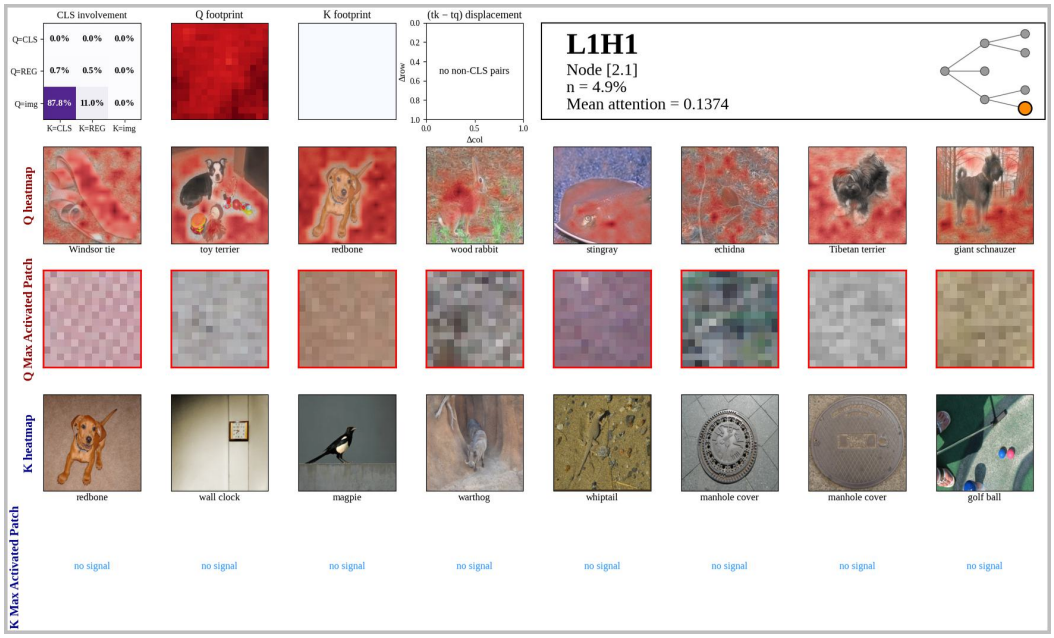
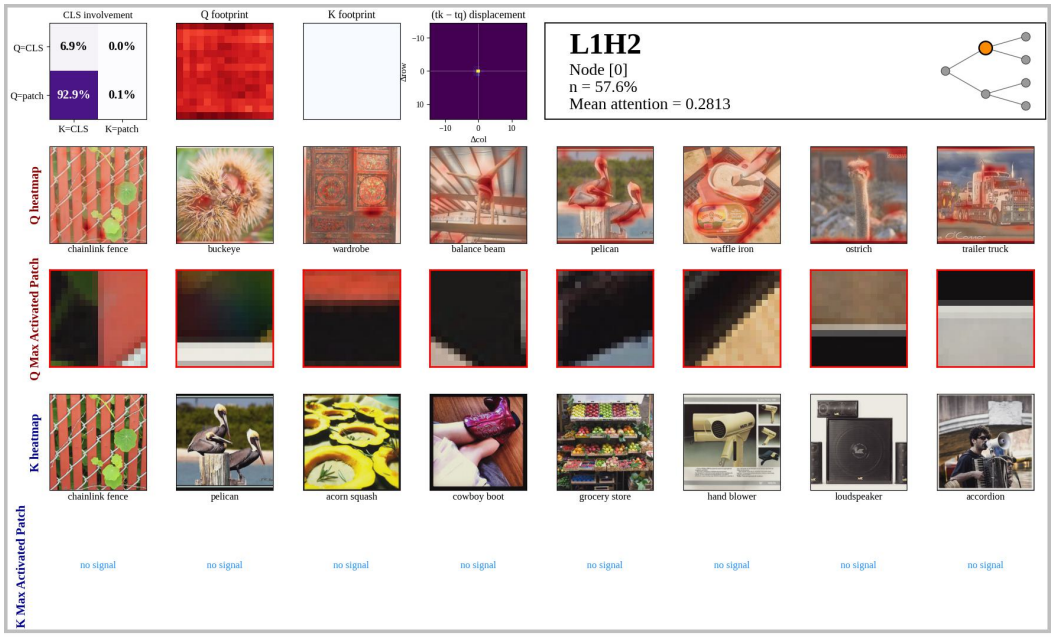


Figure 48: Example role card of DINOv2-L1H1 (node [2.1]).



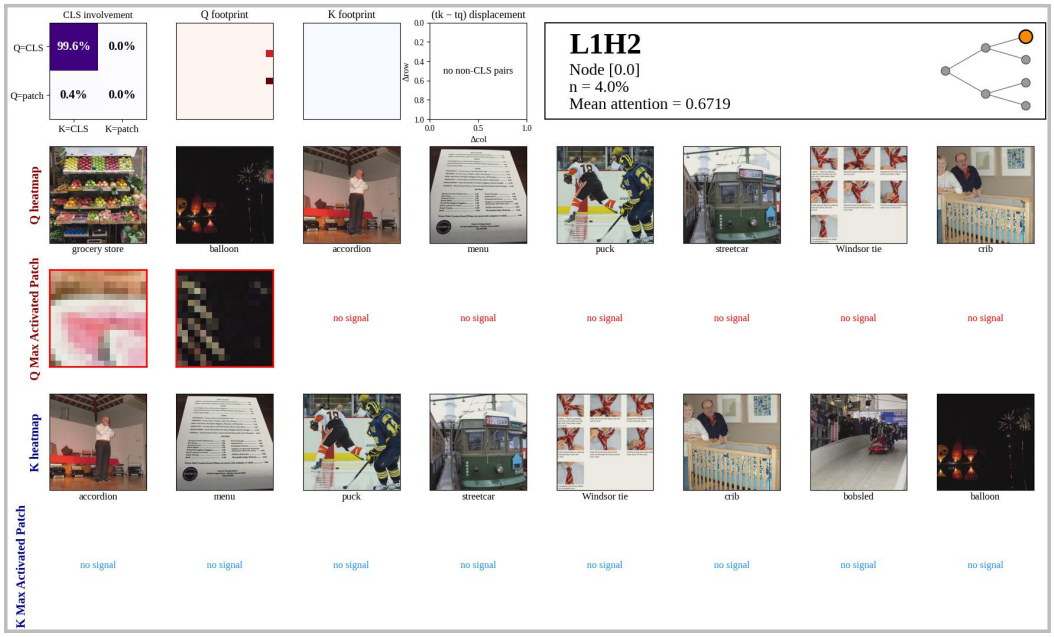


Figure 49: Example role cards of CLIP-L1H2 (nodes [0] and [0.0]).

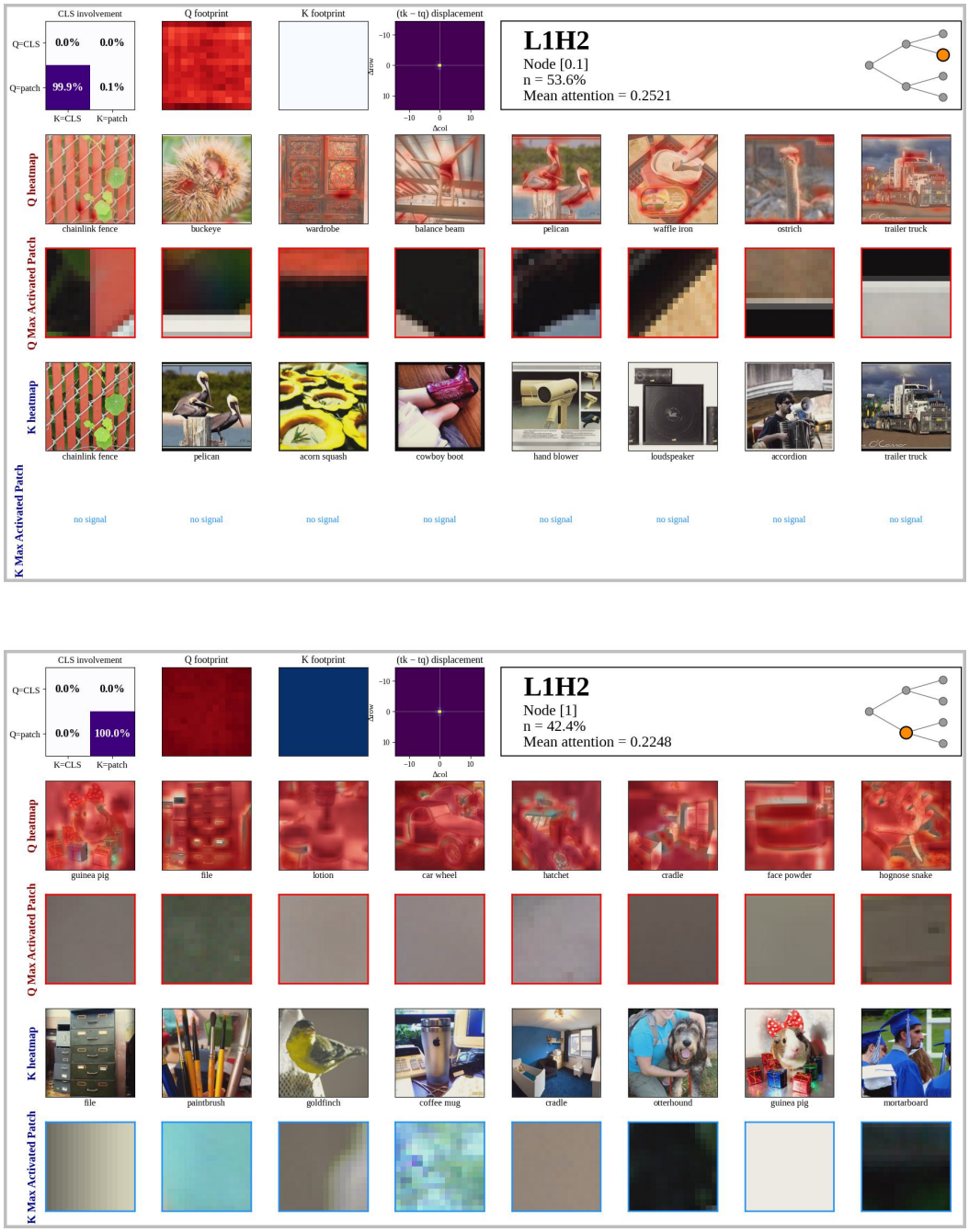
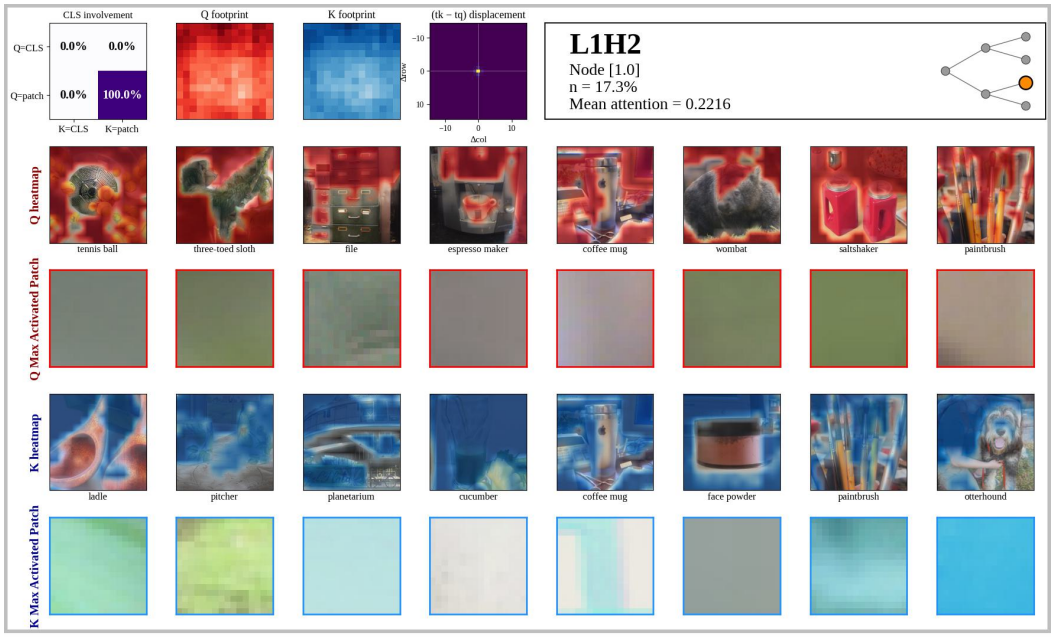


Figure 50: Example role cards of CLIP-L1H2 (nodes [0.1] and [1]).



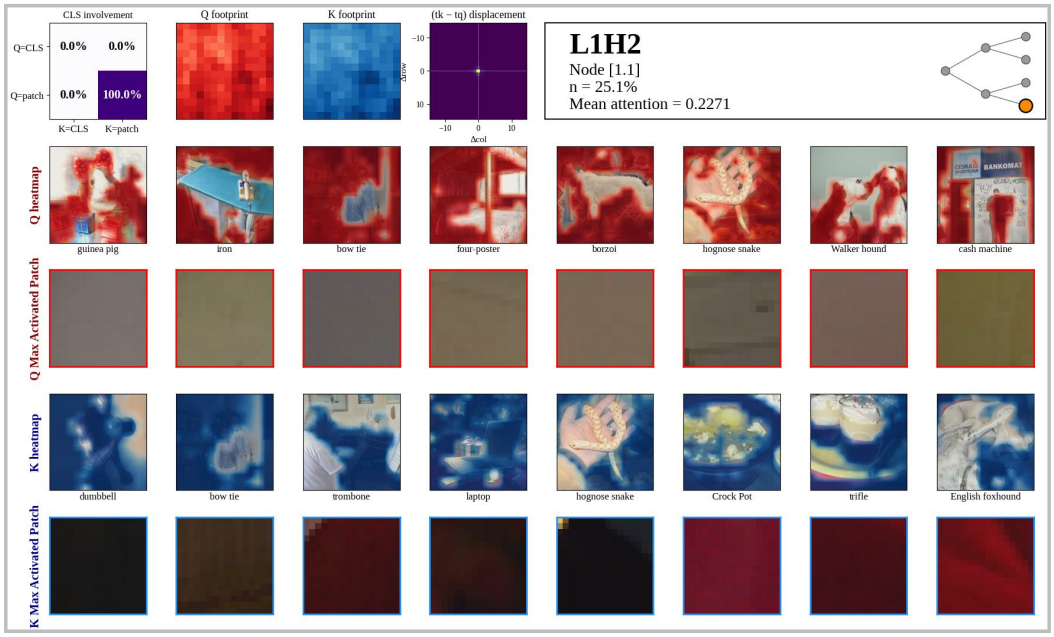


Figure 51: Example role cards of CLIP-L1H2 (nodes [1.0] and [1.1]).

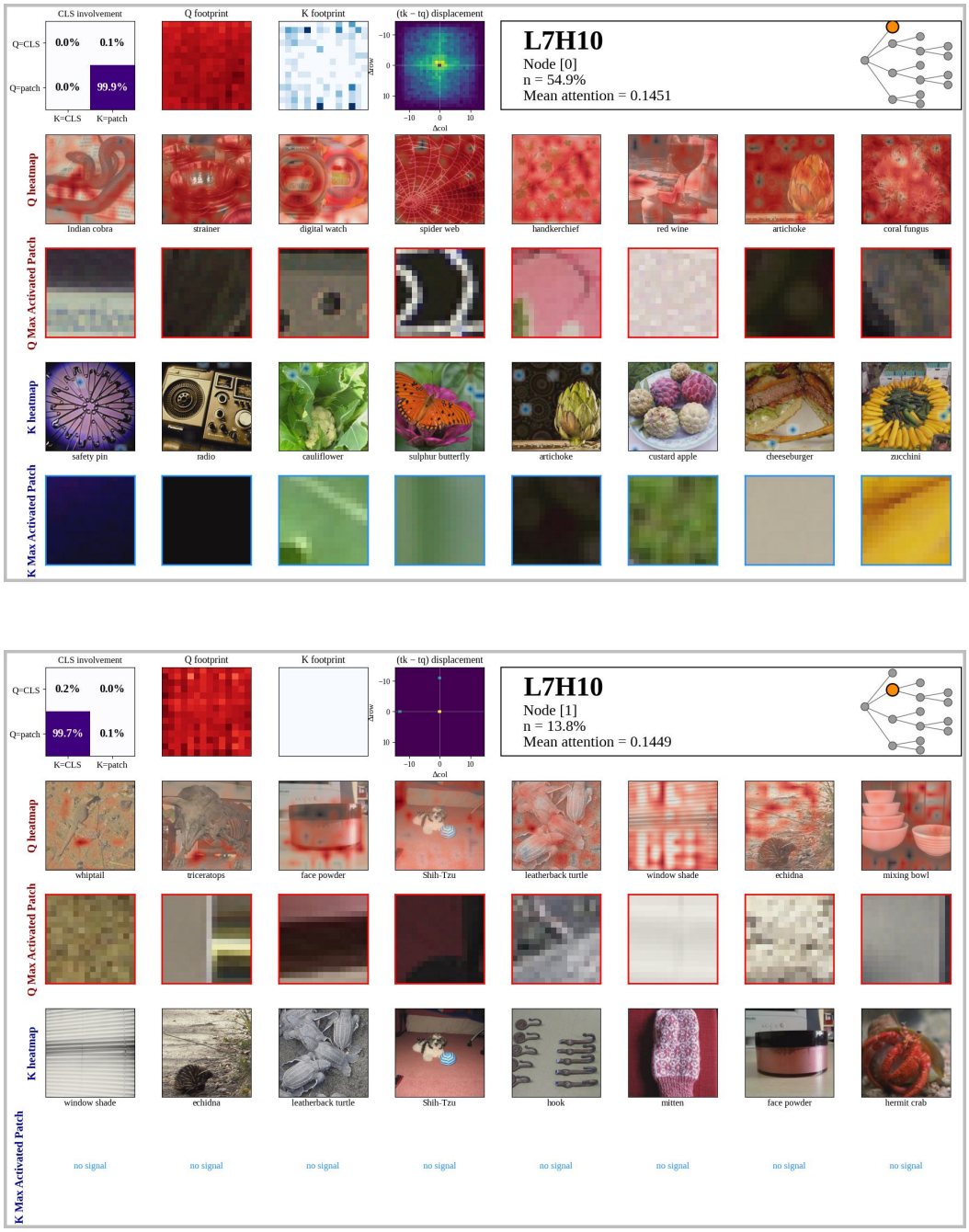
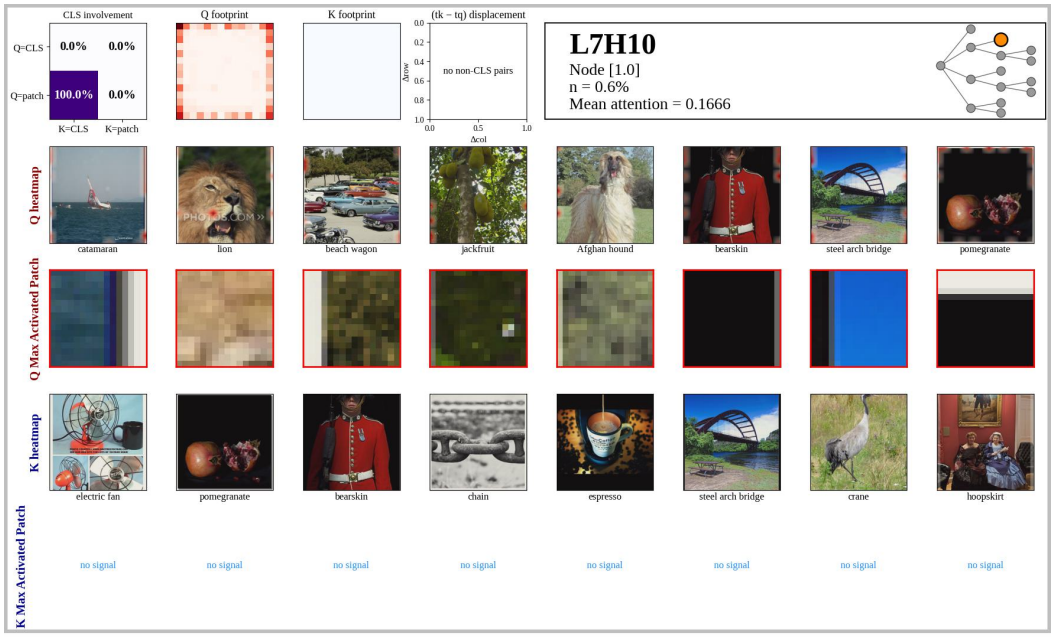


Figure 52: Example role cards of CLIP-L7H10 (nodes [0] and [1]).



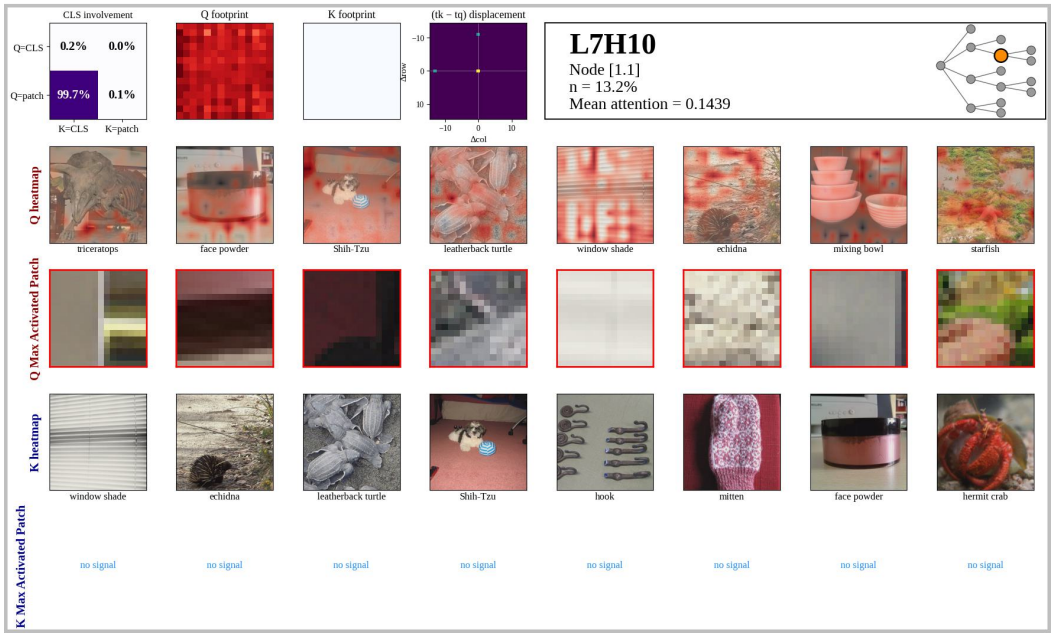


Figure 53: Example role cards of CLIP-L7H10 (nodes [1.0] and [1.1]).



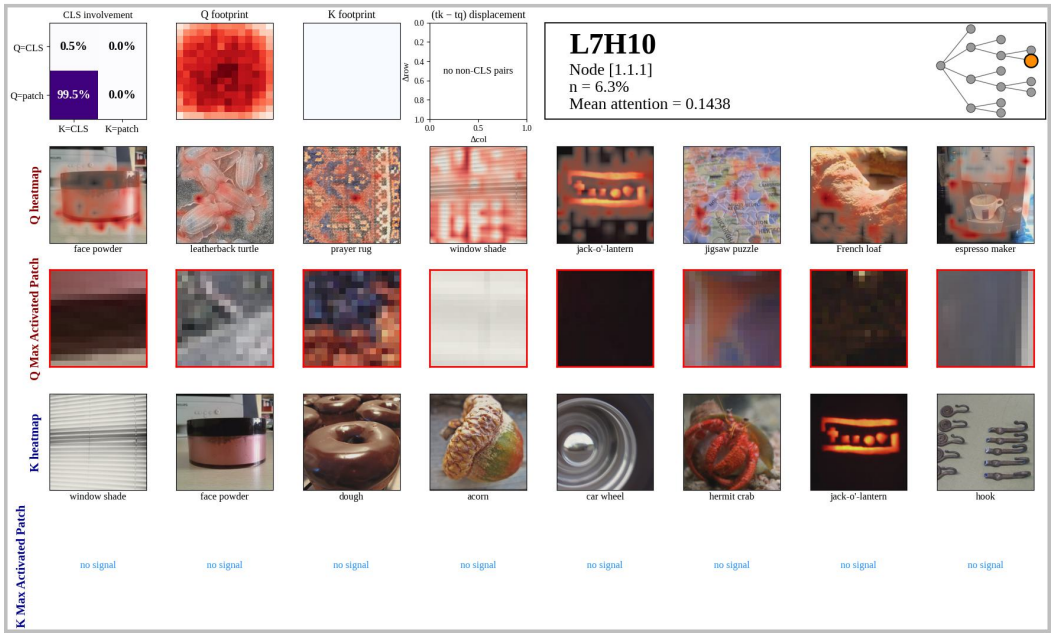


Figure 54: Example role cards of CLIP-L7H10 (nodes [1.1.0] and [1.1.1]).

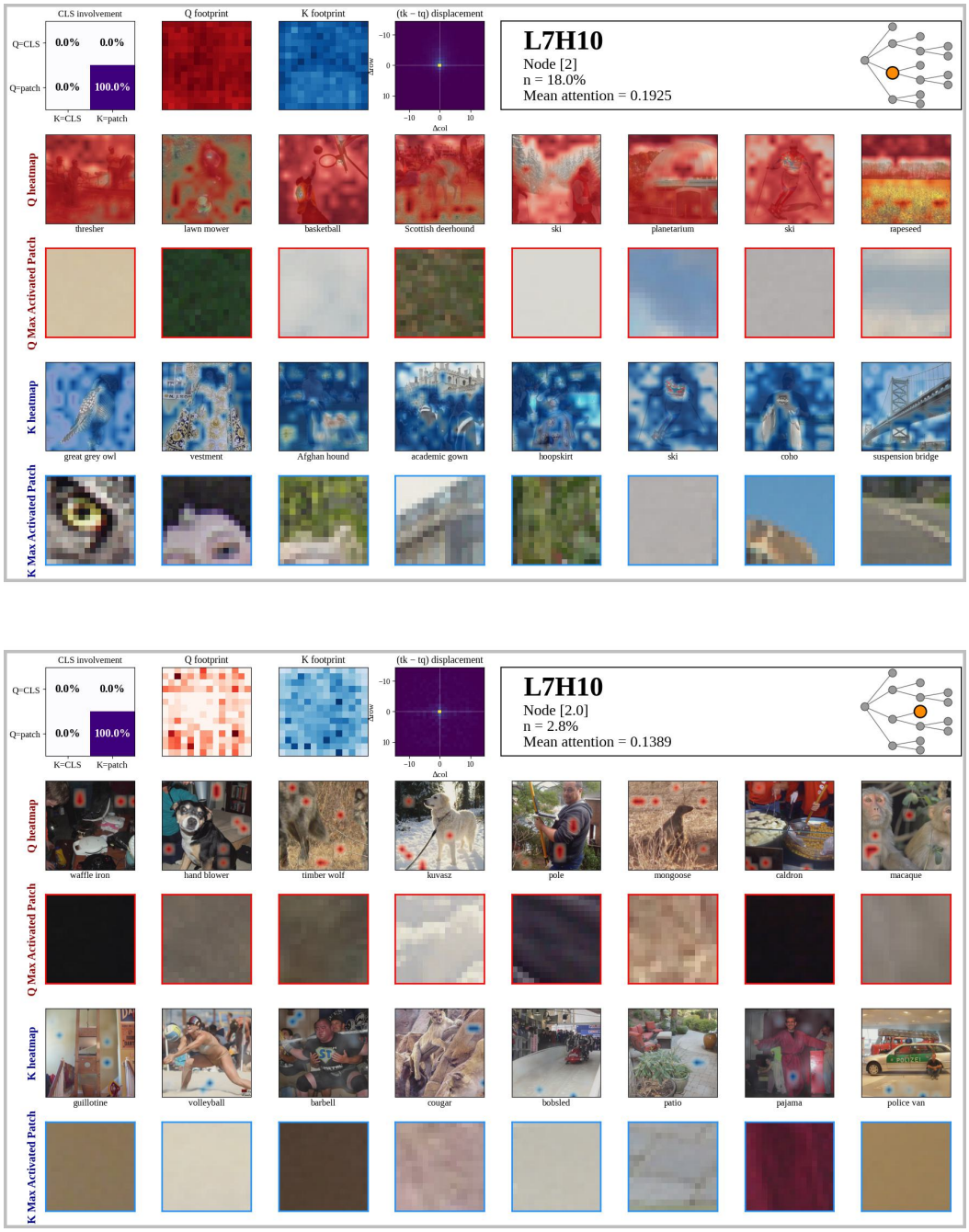
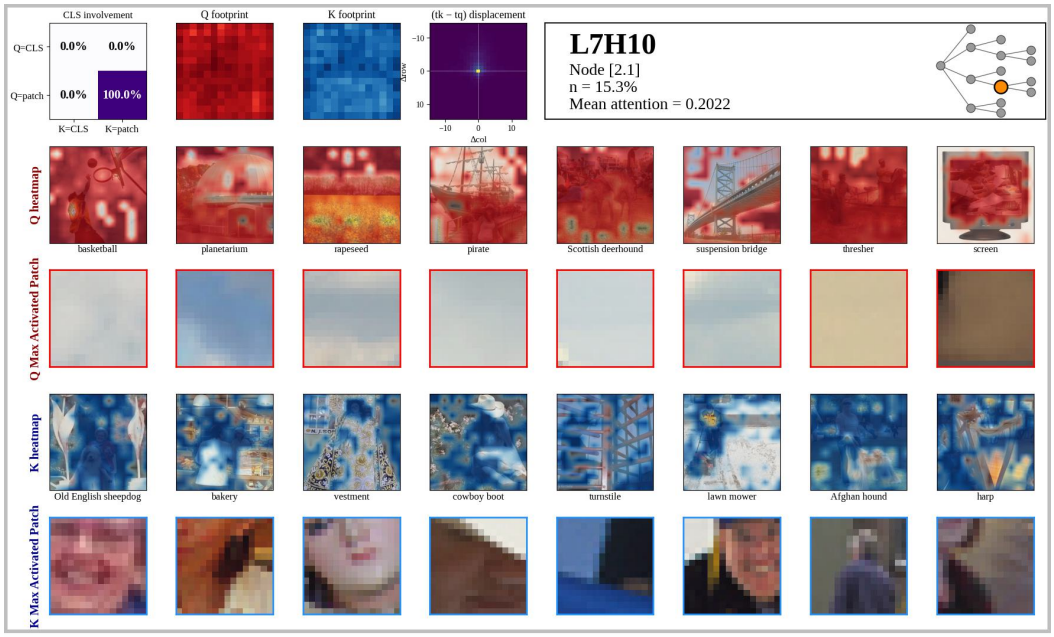


Figure 55: Example role cards of CLIP-L7H10 (nodes [2] and [2.0]).



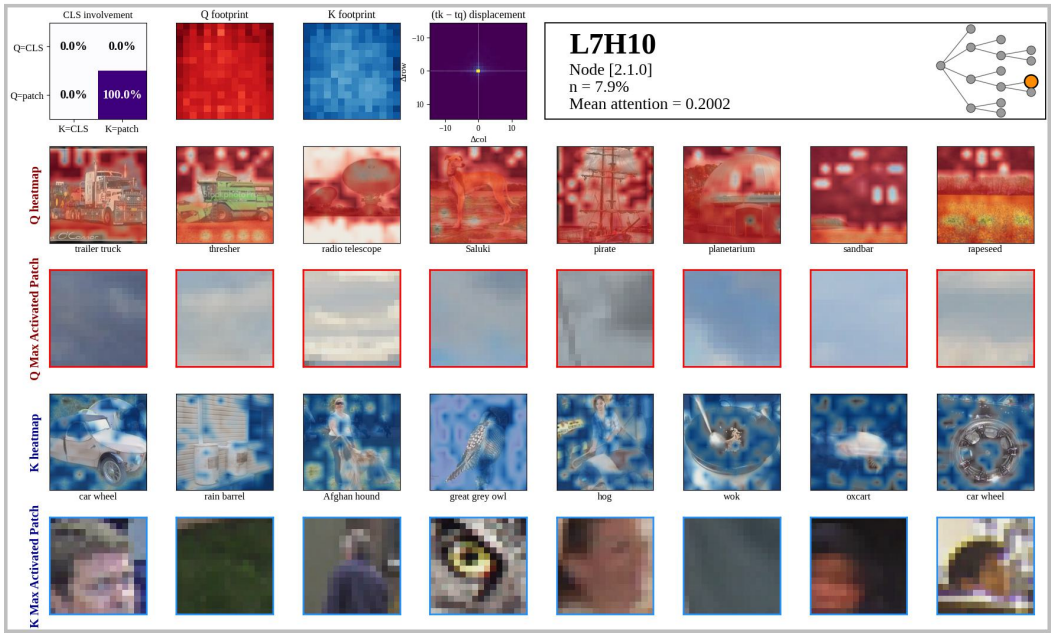


Figure 56: Example role cards of CLIP-L7H10 (nodes [2.1] and [2.1.0]).



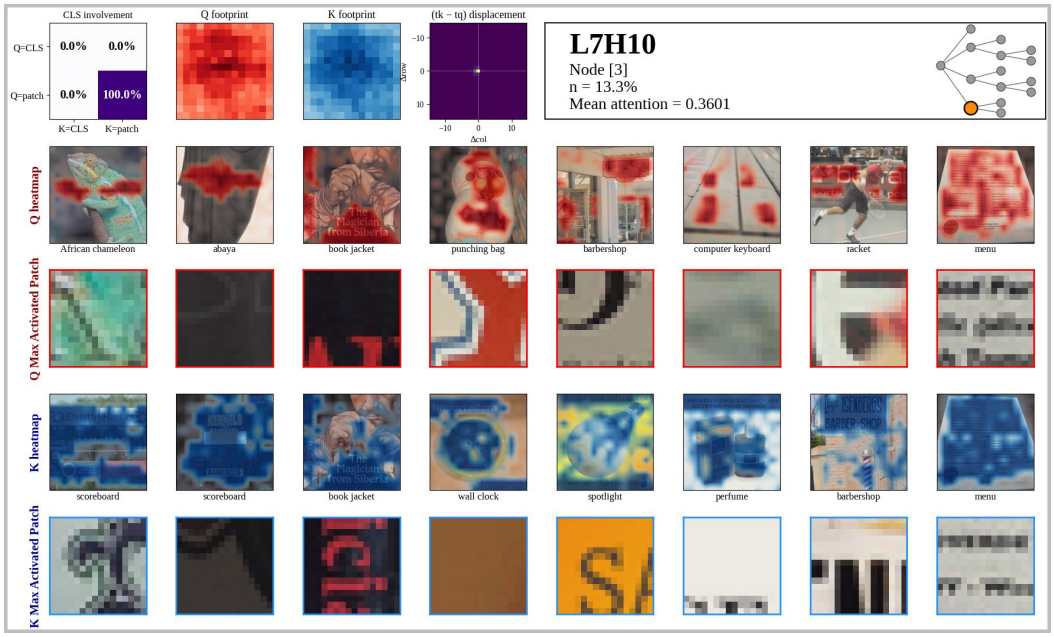


Figure 57: Example role cards of CLIP-L7H10 (nodes [2.1.1] and [3]).

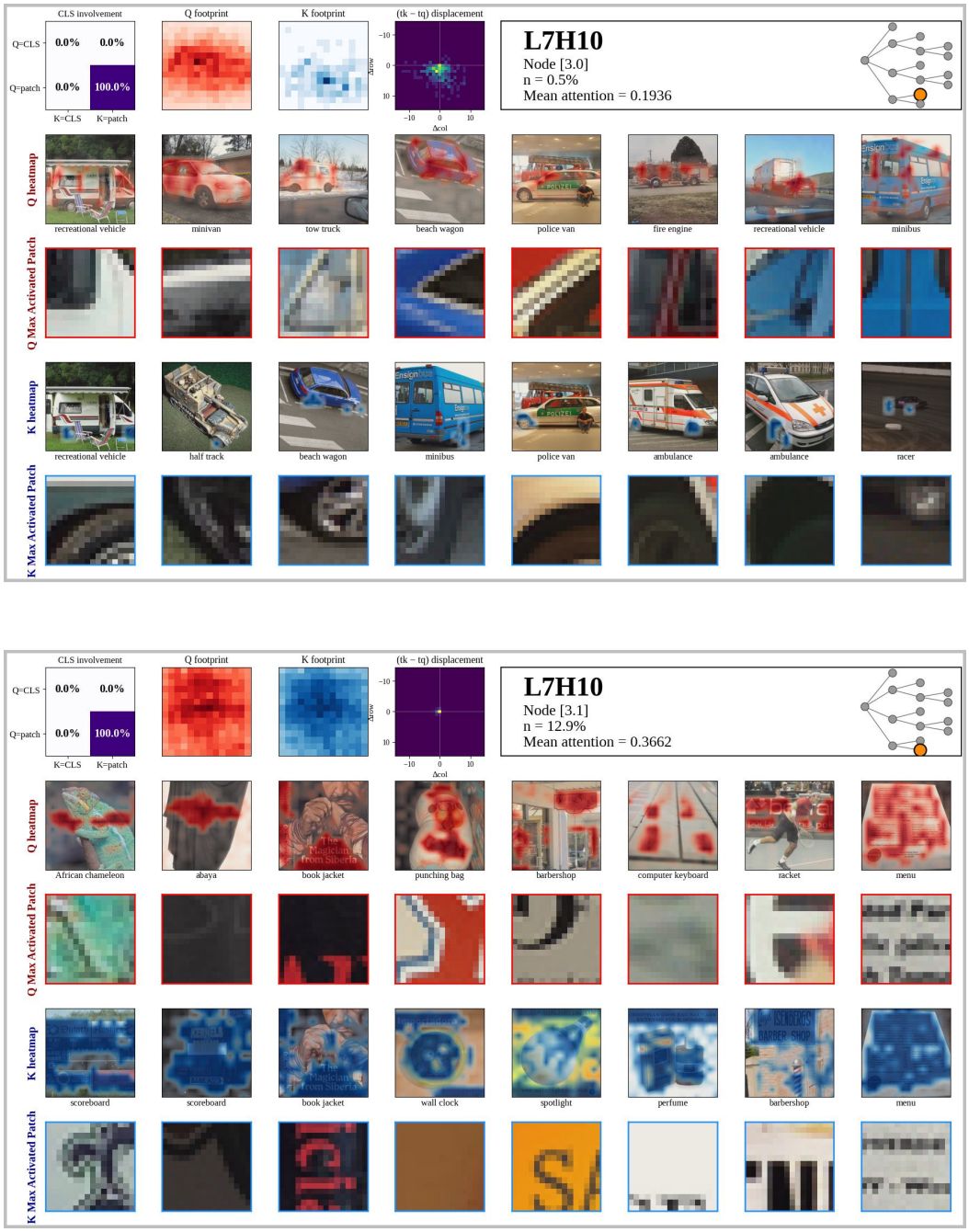


Figure 58: Example role cards of CLIP-L7H10 (nodes [3.0] and [3.1]).

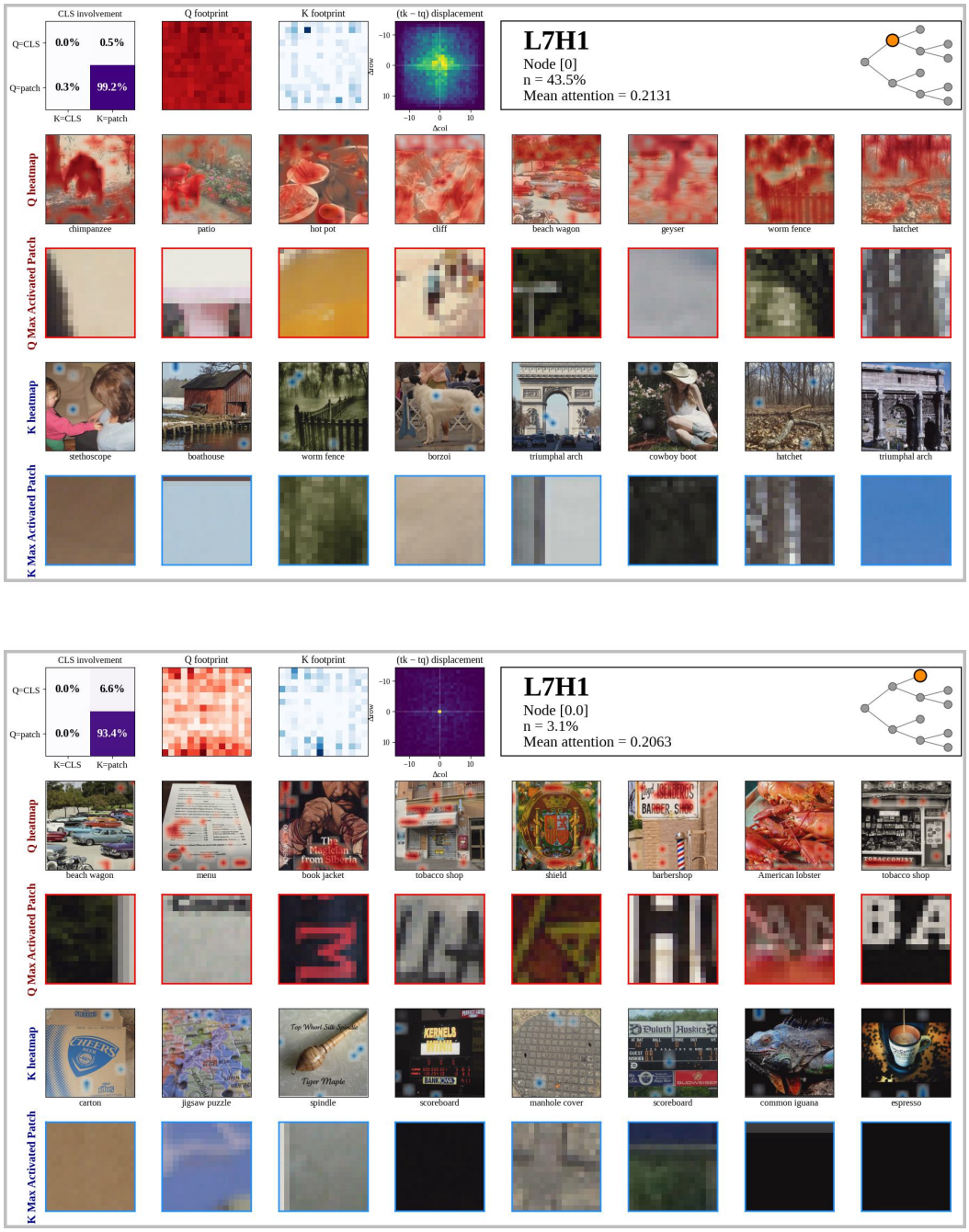
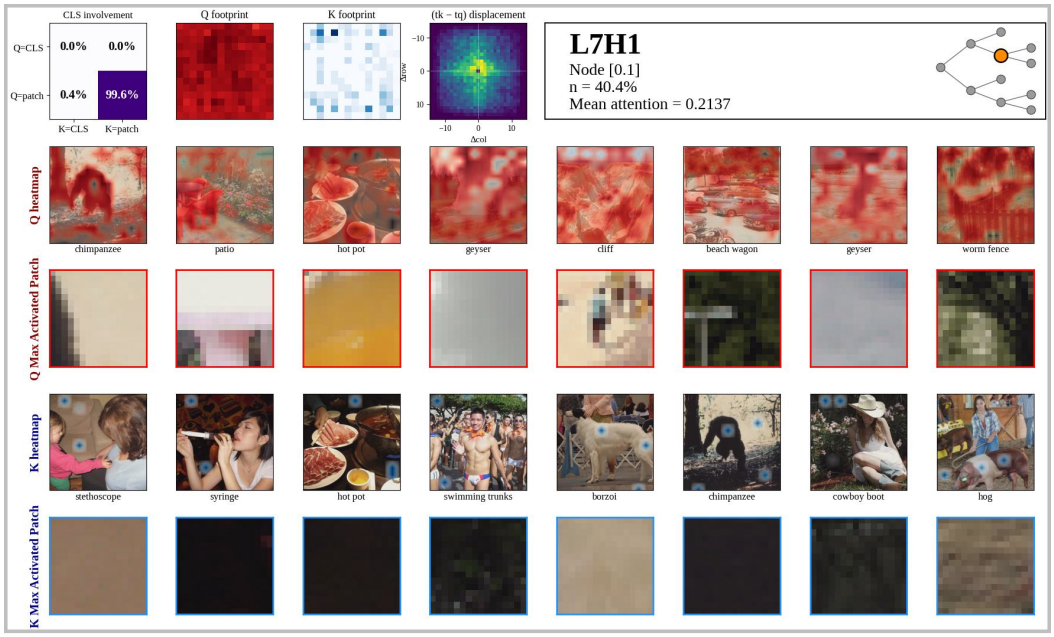


Figure 59: Example role cards of CLIP-L7H1 (nodes [0] and [0.0]).



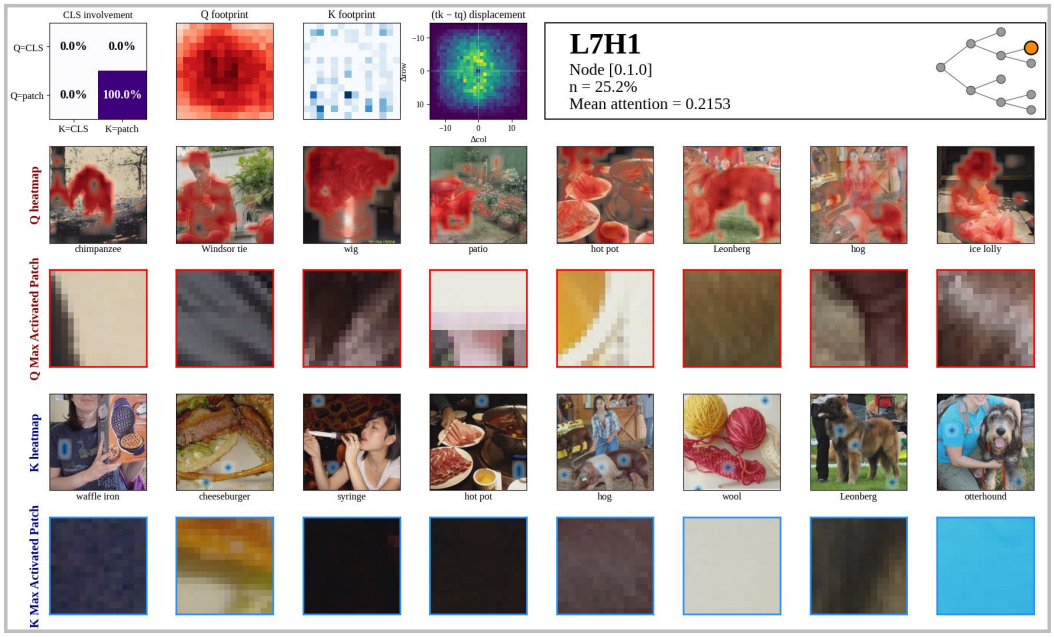
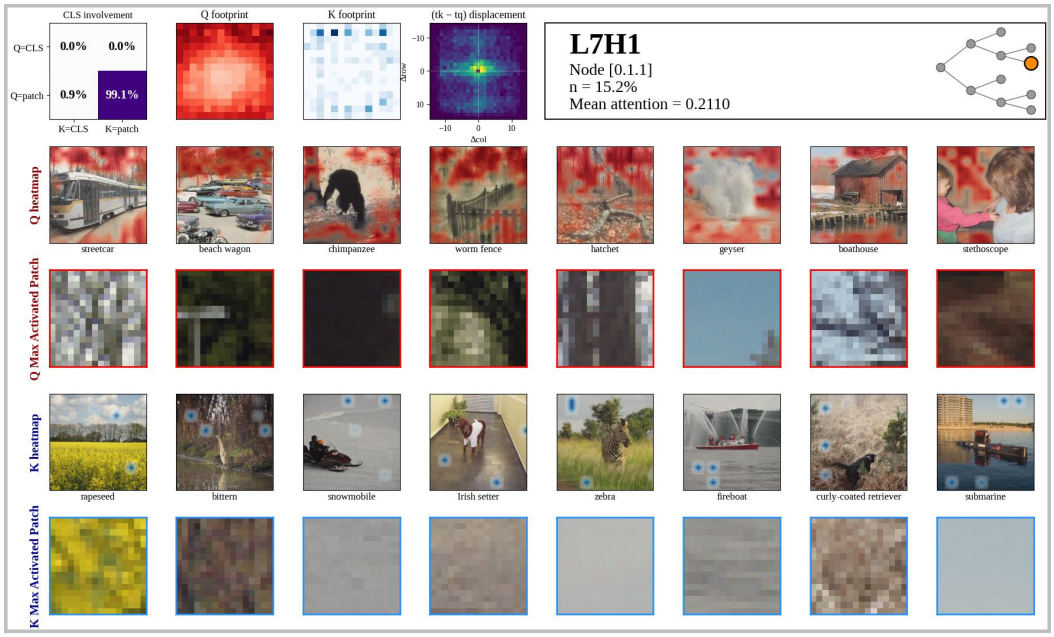


Figure 60: Example role cards of CLIP-L7H1 (nodes [0.1] and [0.1.0]).



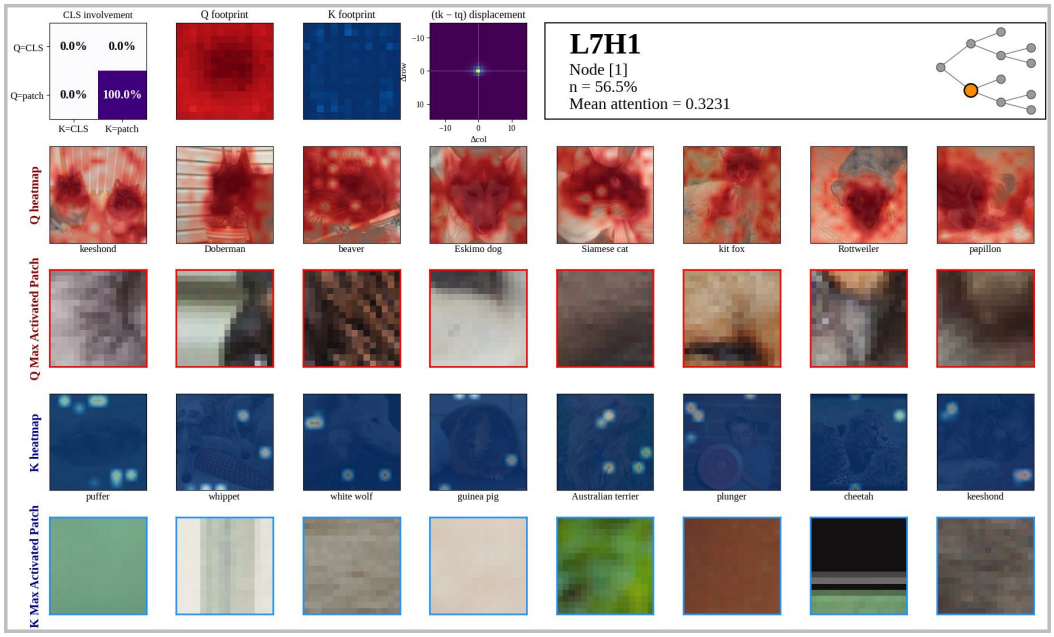
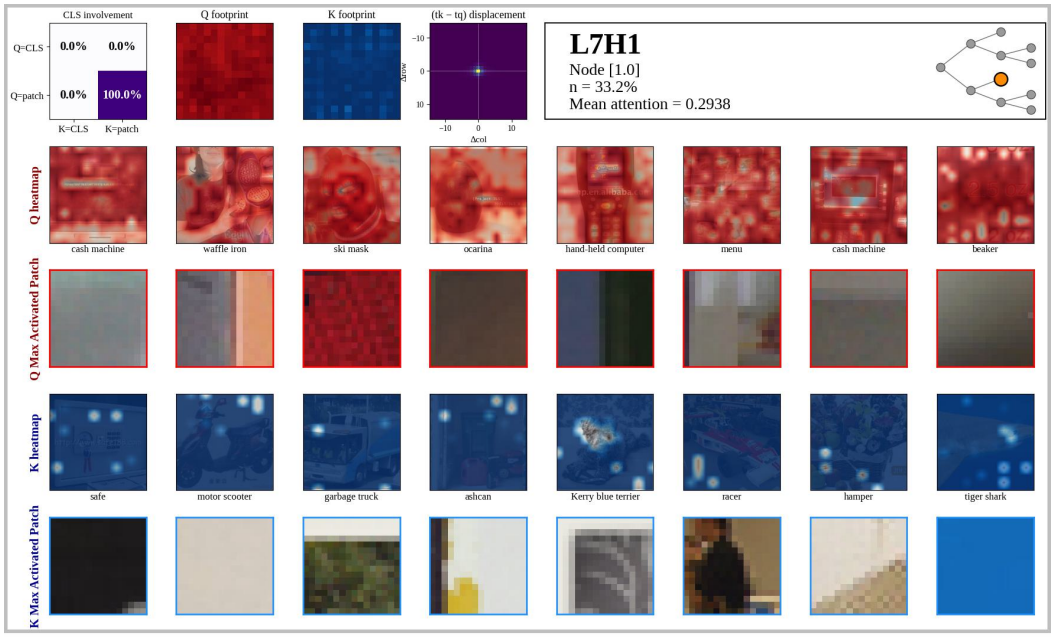


Figure 61: Example role cards of CLIP-L7H1 (nodes [0.1.1] and [1]).



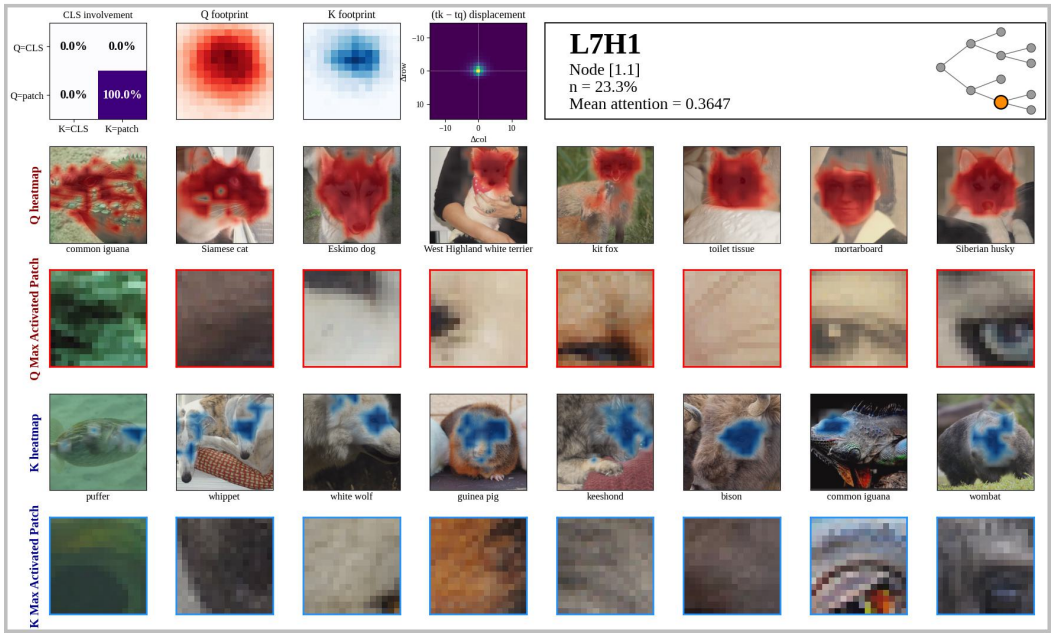


Figure 62: Example role cards of CLIP-L7H1 (nodes [1.0] and [1.1]).

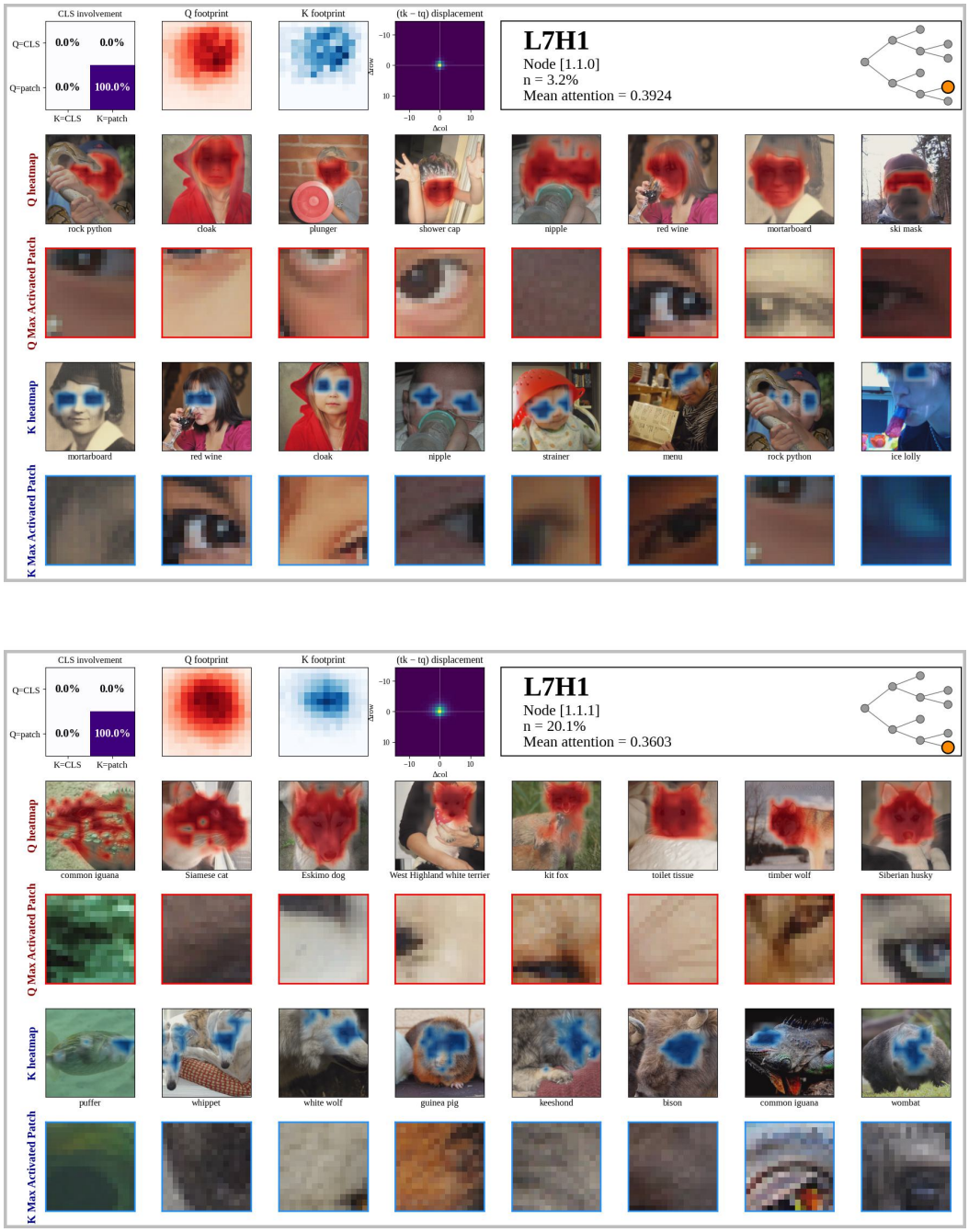


Figure 63: Example role cards of CLIP-L7H1 (nodes [1.1.0] and [1.1.1]).

I Positional Structure in Early-Layer QK Trees

I.1 Positional-leaf Criterion and Coverage Metrics

We adapt the positional-neuron diagnostic of Voita et al. (2024) from individual MLP units to QK tree leaves. For each backbone, layer $\ell \in \{0, 1, 2\}$, head h, leaf v, and side $s \in \{\mathrm{Q}, \mathrm{K}\}$, we apply the following procedure.

1. We select the $N_{\mathrm{cand}} = 50$ candidate images for v using the Stage-1 procedure of Appendix C.5, namely the top 50 images by cluster-pair count. For each candidate image I, we compute the all-regime attention-weighted heatmap $h^s_{I,v} \in \mathbb{R}^{P \times P}$ from Equation (9).
2. We binarize each heatmap relative to its per-image maximum,

$$\mathrm{act}^s_{I,v}(c) \;=\; \mathbb{1}\left[h^s_{I,v}(c) \;\geq\; \tfrac{1}{2} \max_{c'} h^s_{I,v}(c')\right]. \tag{15}$$

   This yields a binary activation indicator analogous to the ReLU > 0 criterion of Voita et al. (2024), adapted to non-negative heatmaps through relative thresholding. If $\max_{c'} h^s_{I,v}(c') = 0$, we set $\mathrm{act}^s_{I,v}(c) = 0$ for all c.
3. Stacking these indicators across the N_{cand} candidate images yields an activation matrix of shape (N_{cand}, P^2). We then compute the mutual-information criterion of Voita et al. (2024) between activation and spatial position,

$$\mathrm{MI}^s_v \;=\; \frac{1}{P^2} \sum_c \left[\mathrm{fr}^s_v(c) \log \frac{\mathrm{fr}^s_v(c)}{\mathrm{fr}^s_v} \;+\; \big(1 - \mathrm{fr}^s_v(c)\big) \log \frac{1 - \mathrm{fr}^s_v(c)}{1 - \mathrm{fr}^s_v}\right], \tag{16}$$

   where $\mathrm{fr}^s_v(c) = \frac{1}{N_{\mathrm{cand}}} \sum_{I \in \mathcal{P}_v} \mathrm{act}^s_{I,v}(c)$ is the activation rate at cell c and $\mathrm{fr}^s_v = \frac{1}{P^2} \sum_c \mathrm{fr}^s_v(c)$ is the global activation rate. We use the convention $0 \log(0/x) = 0$.
4. Following Voita et al. (2024), we classify leaf v as *positional on side* s when $\mathrm{MI}^s_v > 0.05$. A larger value indicates a stronger dependence of the leaf's activation on spatial position.

We summarize positional structure at two levels. The *head-side union* is $U^s_{\ell,h}(c) = \max_{v \in \mathcal{T}^{\ell,h},\, \mathrm{MI}^s_v > 0.05} \mathrm{fr}^s_v(c)$, the per-cell maximum over all positional leaves on side s within head (ℓ, h). The *layer-side union* is defined analogously by taking the maximum over all positional leaves in layer ℓ, regardless of head. When a head or layer contains no positional leaf on side s, we define the corresponding union to be zero at every cell. We define the *coverage* of a union heatmap at threshold $\tau = 0.5$ as the fraction of grid cells whose union value exceeds τ. This quantity measures the fraction of the $P \times P$ patch grid covered by positional leaves.

I.2 Experimental Setup for Positional Analysis

For the positional analysis, we modify the main-paper QK tree configuration in two ways. All remaining hyperparameters, including UMAP, HDBSCAN, feature encoding, the branching cap, the classifier, and $d_{\max} = 3$, follow Appendix C. The relaxed configurations introduced here are used only for the positional analysis in this section. The cluster-consistency results of §5.2, including the depth progression of $\bar{r}^s$, are computed with the original main-paper trees.

(1) Silhouette gate $s_{\mathrm{gate}} = 0.3$ instead of 0.5. The main analysis uses $s_{\mathrm{gate}} = 0.5$, a conventional threshold for reasonably separated cluster structure following Rousseeuw (1987). This conservative setting is appropriate for the semantic-consistency analysis in §5.2, where we prioritize clearly separated subclusters. For the positional analysis, however, weaker spatial partitions can fall below this threshold. In early-layer heads, we observe coarse patterns that divide the image into broad spatial regions, while their silhouette scores often lie between 0.30 and 0.45. We therefore rebuild the QK trees for layers $\ell \in \{0, 1, 2\}$ of all four backbones using $s_{\mathrm{gate}} = 0.3$, while keeping the remaining clustering hyperparameters unchanged. The relaxed gate increases the number of heads containing positional leaves. For example, among the 36 early-layer heads of supervised ViT, 19 contain no positional leaves under $s_{\mathrm{gate}} = 0.5$. After lowering the gate, 12 of these heads acquire positional leaves. One previously positional head no longer contains a positional leaf, while the remaining 7 heads are non-positional under both settings.

(2) Restricting the analysis to layers $\ell \in \{0, 1, 2\}$. Our sweep across layers shows that positional mutual information is concentrated primarily in the early layers. For nearly all leaves beyond layer $\ell = 2$, the criterion in Equation (16) falls below the positional threshold. We therefore restrict the positional analysis and visualizations to the first three layers. Applying the same procedure to deeper layers yields very few positional leaves and little additional coverage.

I.3 Positional Coverage across Backbones

Layer-level positional coverage is nearly complete for ViT and CLIP. Using the relaxed threshold $s_{\text{gate}} = 0.3$, the layer-side union U^s_ℓ covers almost the entire $P \times P$ patch grid for ViT and CLIP across all early layers $\ell \in \{0, 1, 2\}$. At the evaluation threshold $\tau = 0.5$, coverage is 100% in every setting except the query side of CLIP at $\ell = 0$, where it reaches 99.5%. Thus, when positional leaves are aggregated across all 12 heads within an early layer, at least one leaf activates at nearly every patch location, and this near-complete coverage holds on both the query and key sides.

Per-head positional structure is sparse but becomes dense after aggregation. Individual heads typically exhibit positional structure on only one side of the QK interaction. The examples in §I.4 show that many heads contain either Q-positional or K-positional leaves, while relatively few contain both. Accordingly, the positional coverage of a single head is often limited, frequently remaining below 30%. When positional leaves are aggregated across all 12 heads, however, the layer-side union becomes dense. Different heads cover different spatial regions, including edge bands, corners, central patches, and short-range neighborhoods. This resembles the union-style positional decomposition reported by Voita et al. (2024) for positional MLP neurons, but here the structure is recovered from QK-feature clusters rather than neuron activations.

Register-augmented backbones show weaker positional coverage. DINOv2 and DINOv3 exhibit lower positional coverage under the same analysis, particularly on the key side. For DINOv2, full coverage is reached only at $\ell = 1$. At $\ell = 0$, the query-side and key-side unions cover 67% and 56% of the grid, respectively. At $\ell = 2$, key-side coverage drops to 0%. DINOv3 shows an even larger reduction in several early-layer settings. The query/key coverage values are $45\%/5\%$ at $\ell = 0$, $73\%/0\%$ at $\ell = 1$, and $100\%/80\%$ at $\ell = 2$. Across these early layers, both DINOv2 and DINOv3 show lower key-side coverage than query-side coverage.

The reduced DINO-family coverage partly reflects the branching cap. The main-paper trees apply a branching cap of $k_{\max} = 4$ to the register-augmented backbones, as described in Appendix C. Repeating the positional analysis for DINOv2 without the cap raises the layer-side coverage to 100% on both sides at all three early layers, indicating that merging many subclusters into a small number of children can push finer positional partitions below the MI threshold. DINOv3 shows no such recovery. Removing the cap leaves the key-side unions near 0% in the earliest layers, while the query-side union at $\ell = 0$ drops from 45% to 9%. Its early-layer positional structure therefore does not resolve into clusters under the relaxations we tested. We do not attempt to identify the underlying cause, since DINOv3 differs from DINOv2 along several dimensions, including training scale, register-token design, and training procedure.

I.4 Per-head Visualizations of Positional Leaves

For each of the four backbones, we visualize the per-head Q-side and K-side positional leaves for layers $\ell \in \{0, 2\}$. Rows correspond to heads $h \in \{0, \ldots, 11\}$, and only heads containing at least one positional leaf on the corresponding side are shown. Within each row, up to 10 cells show the activation-rate heatmaps fr^s_v of the positional leaves on side s, ranked in decreasing order of MI^s_v. The rightmost red-bordered cell shows the head-side union $U^s_{\ell,h}$. Cell titles of the form "<path> I=<value>" report the leaf path in the QK tree and its positional MI value MI^s_v.

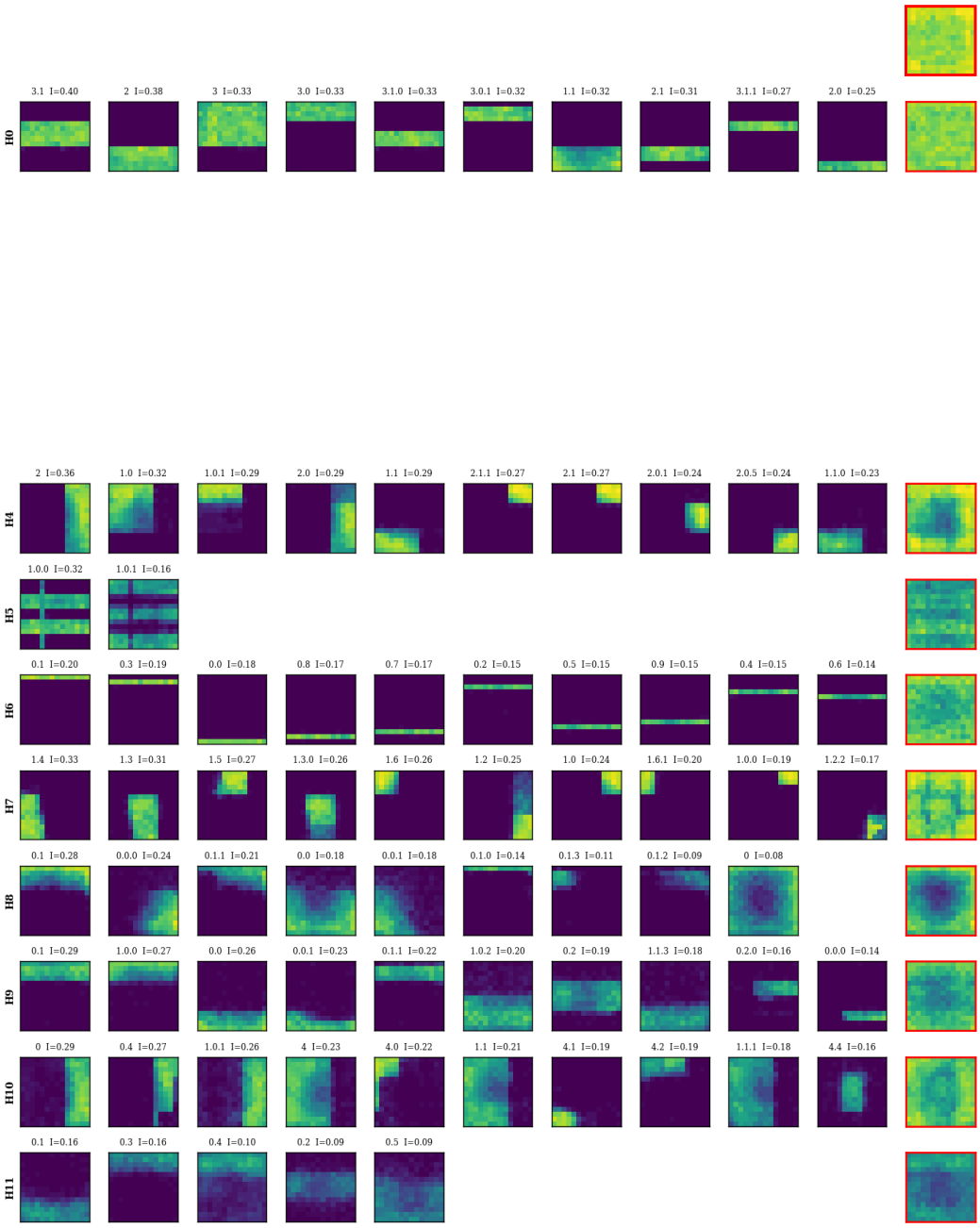


Figure 64: ViT (sup), $\ell = 0$, Q-side positional leaves.

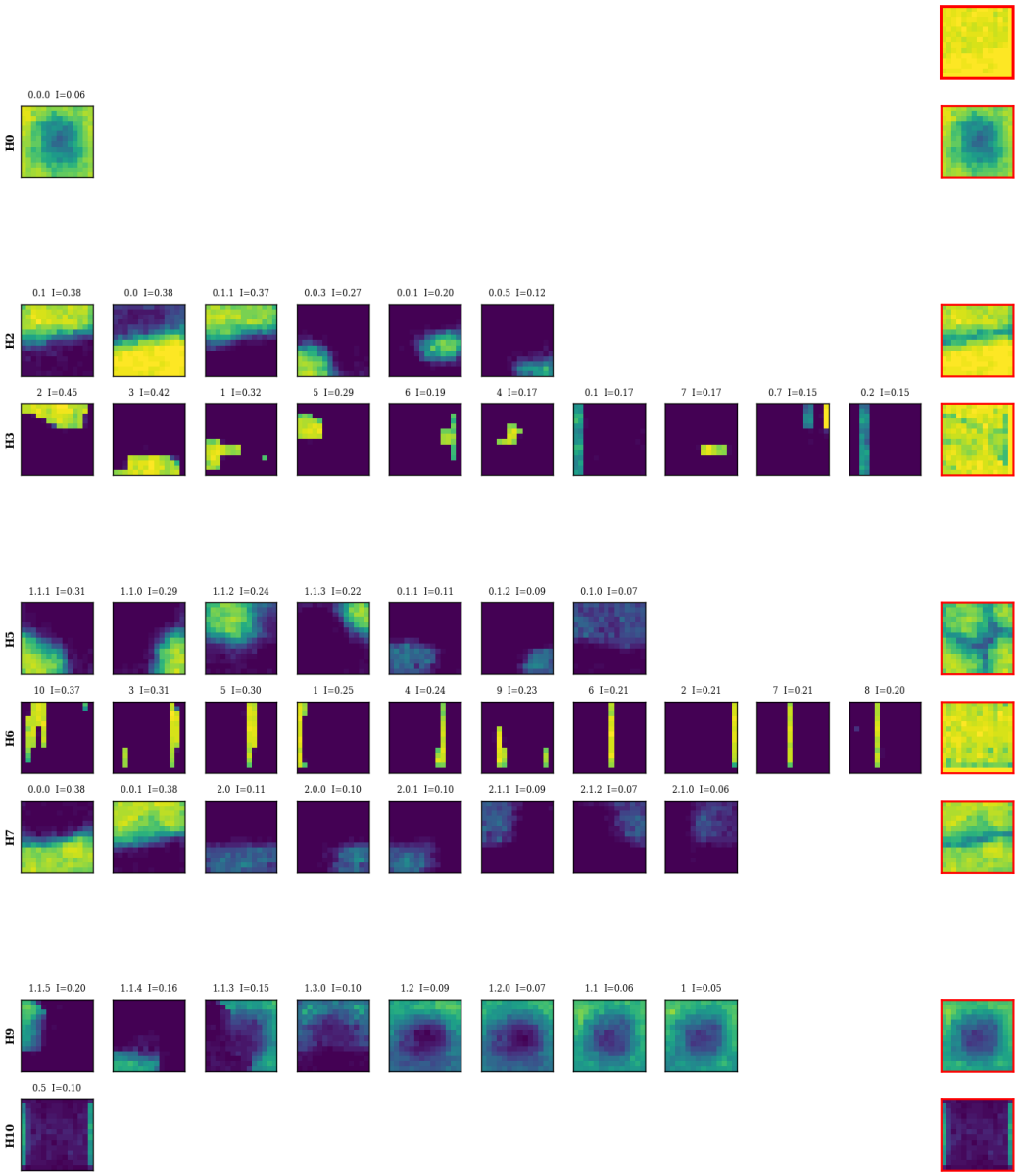


Figure 65: ViT (sup), $\ell = 2$, Q-side positional leaves.

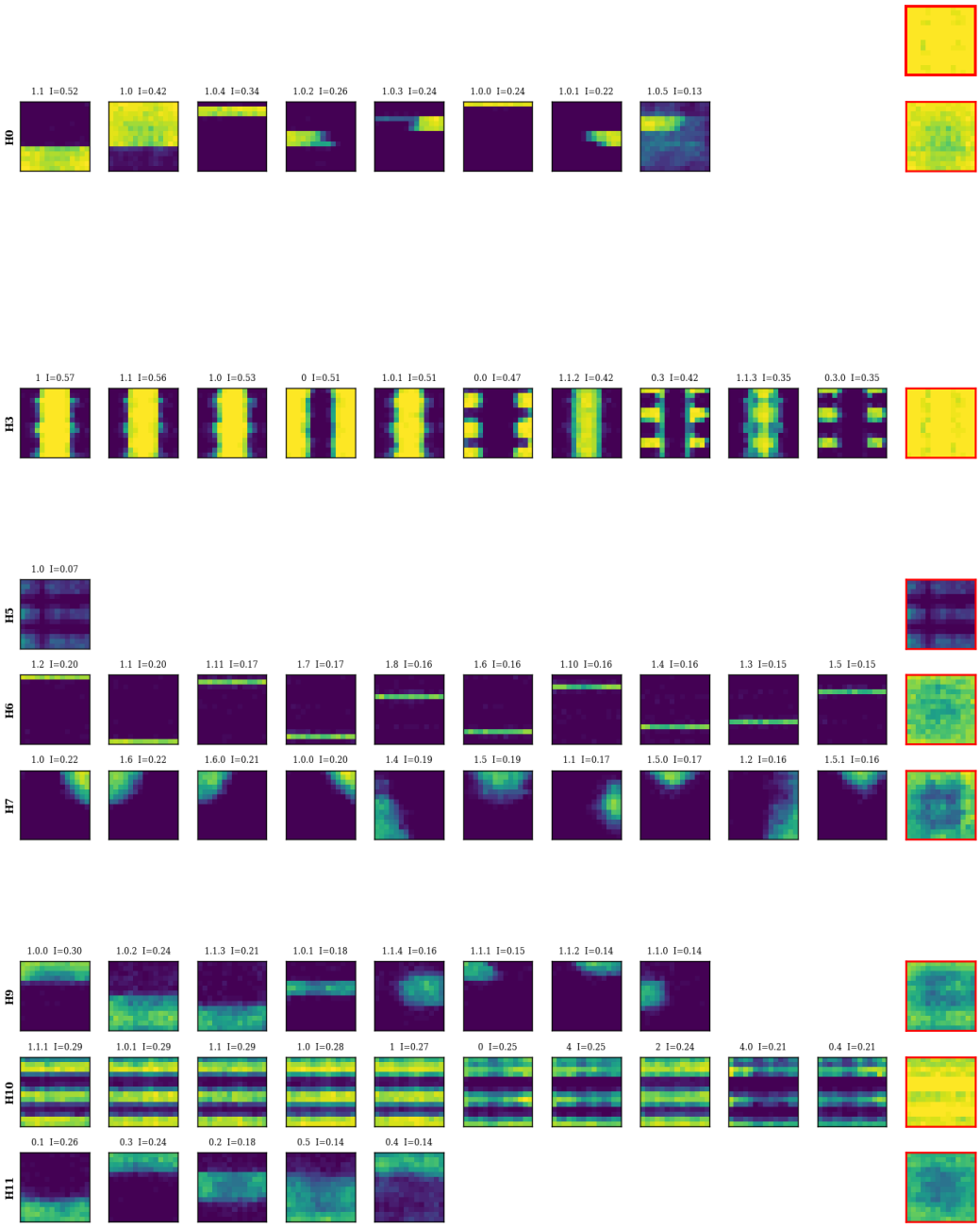


Figure 66: ViT (sup), $\ell = 0$, K-side positional leaves.

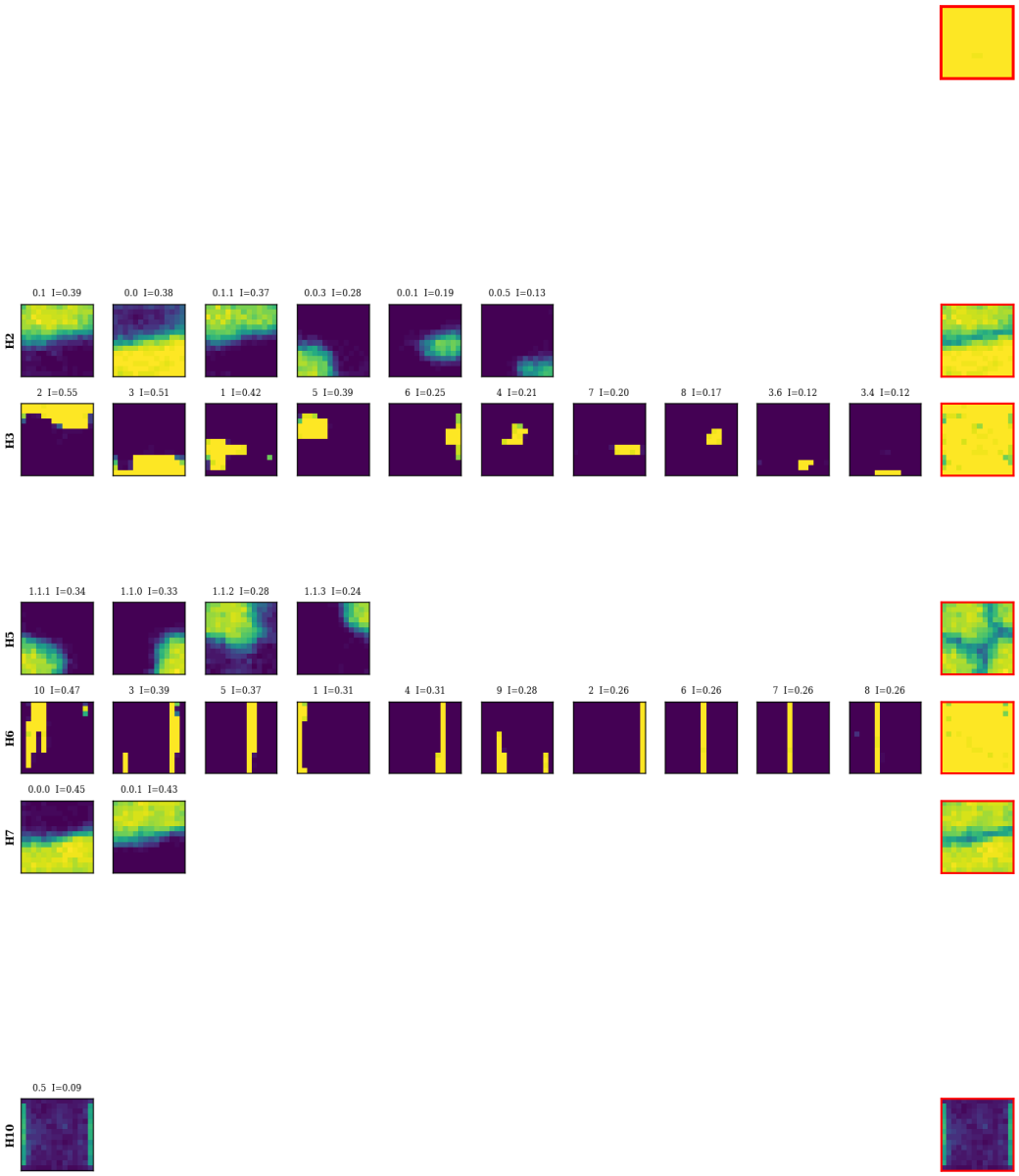


Figure 67: ViT (sup), $\ell = 2$, K-side positional leaves.

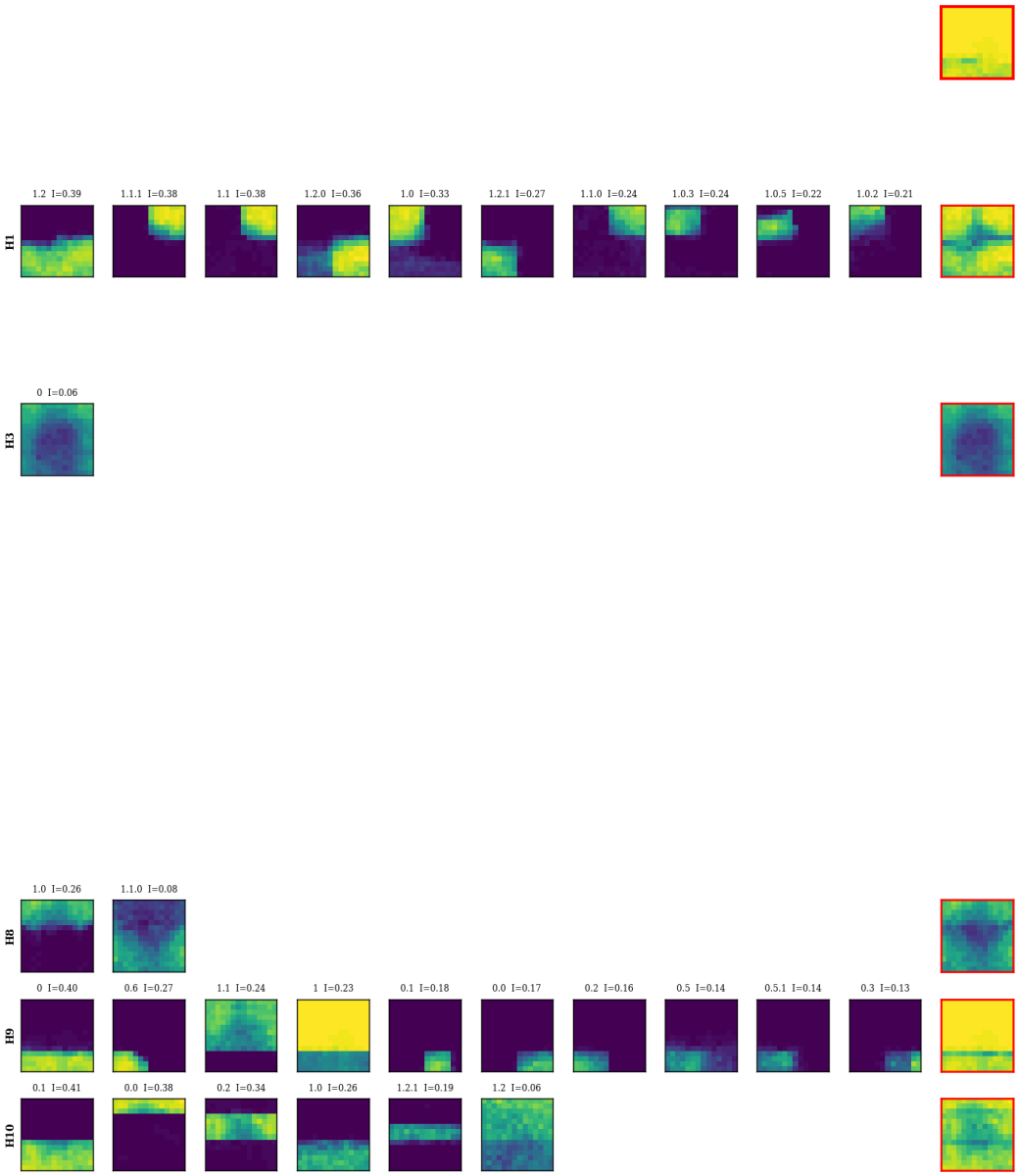


Figure 68: CLIP ViT, $\ell = 0$, Q-side positional leaves.

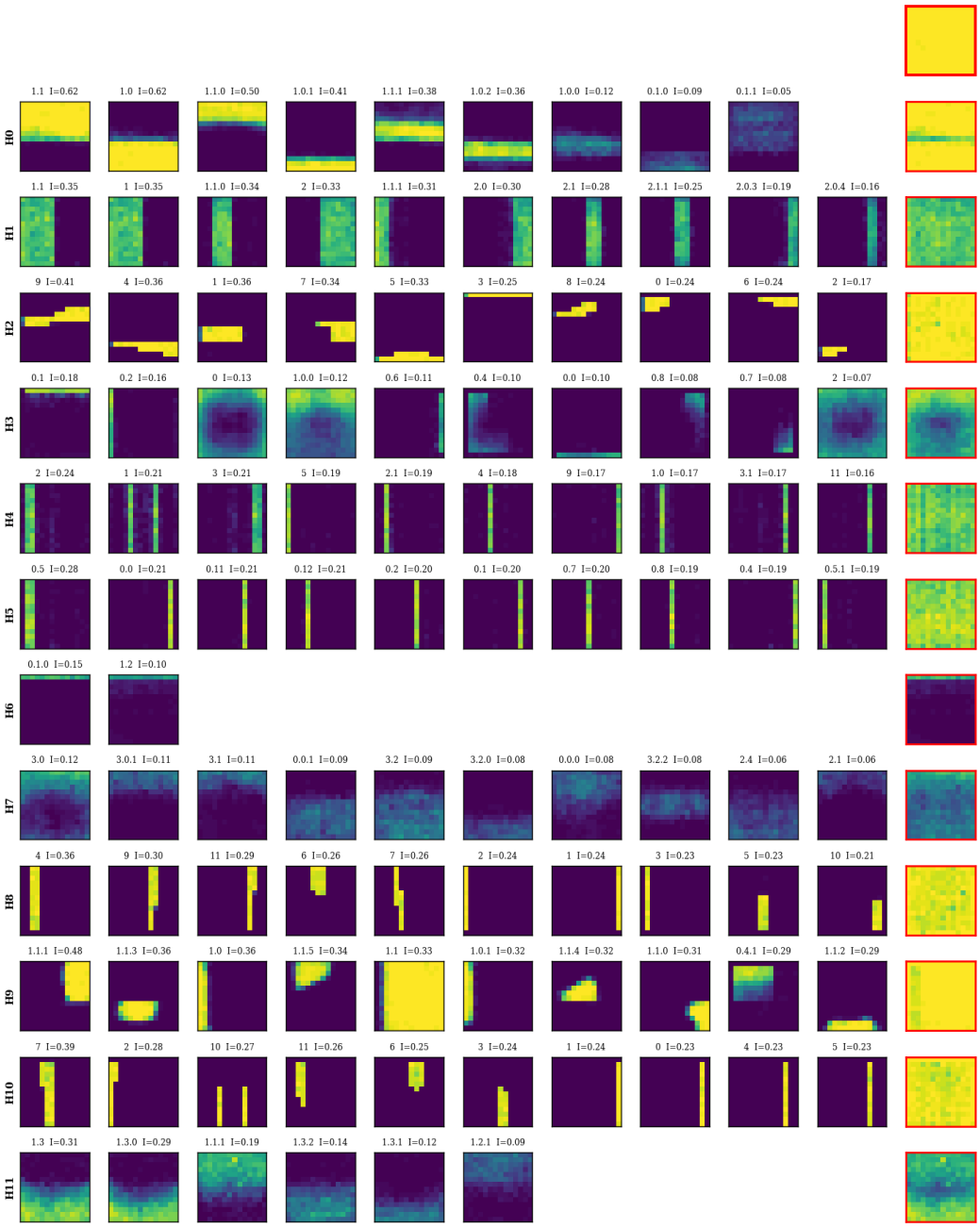


Figure 69: CLIP ViT, $\ell = 2$, Q-side positional leaves.

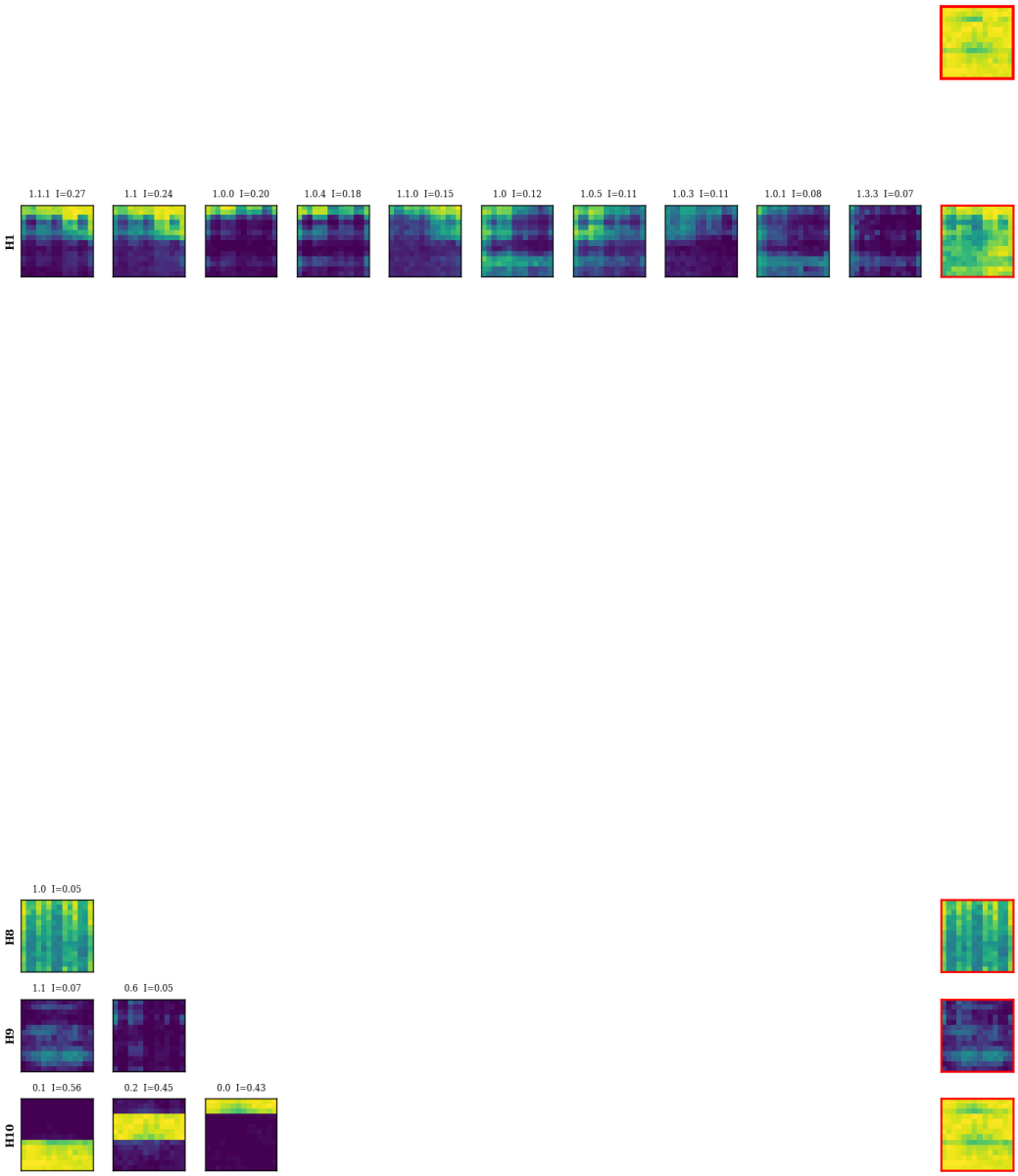


Figure 70: CLIP ViT, $\ell = 0$, K-side positional leaves.

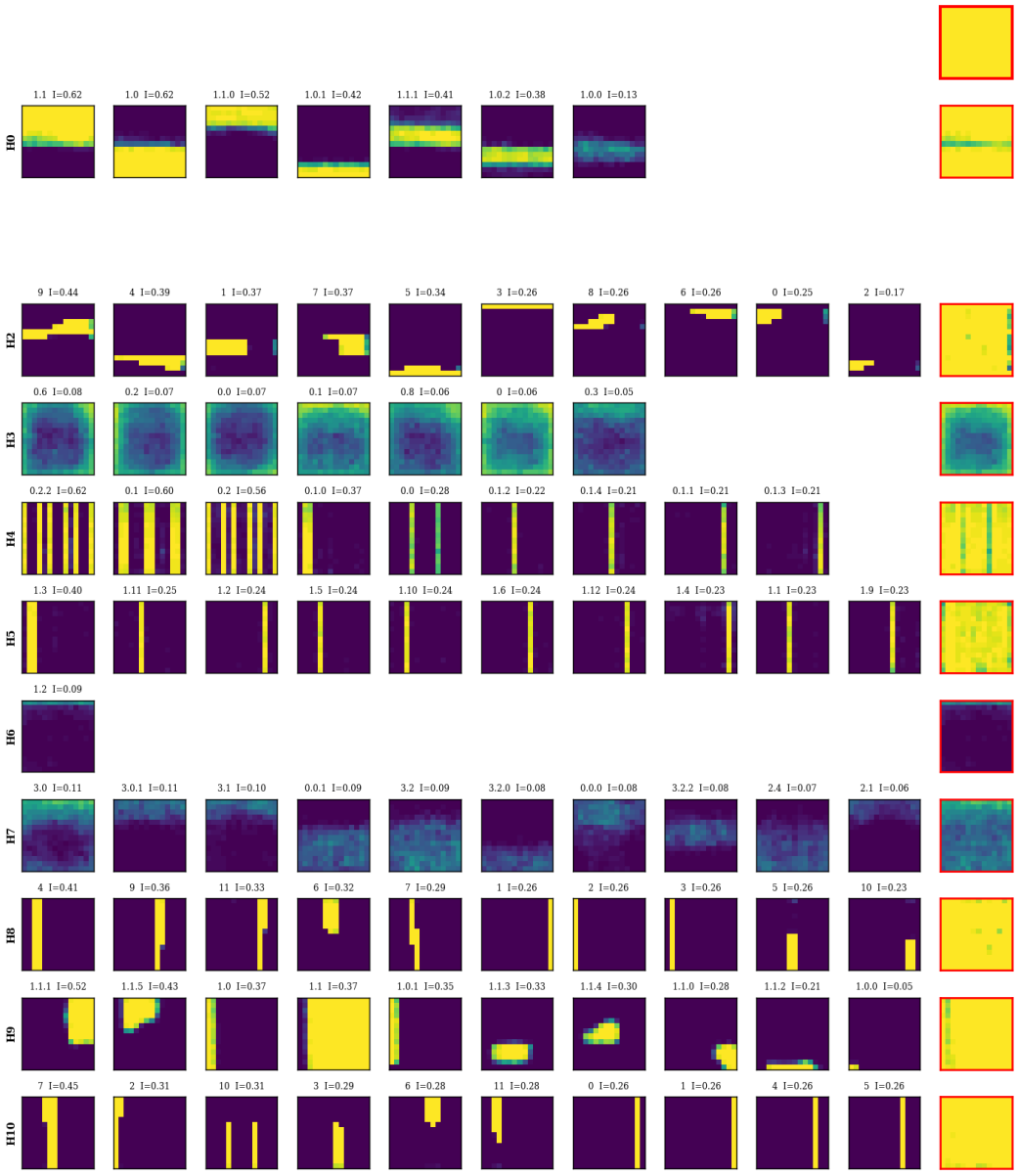


Figure 71: CLIP ViT, $\ell = 2$, K-side positional leaves.

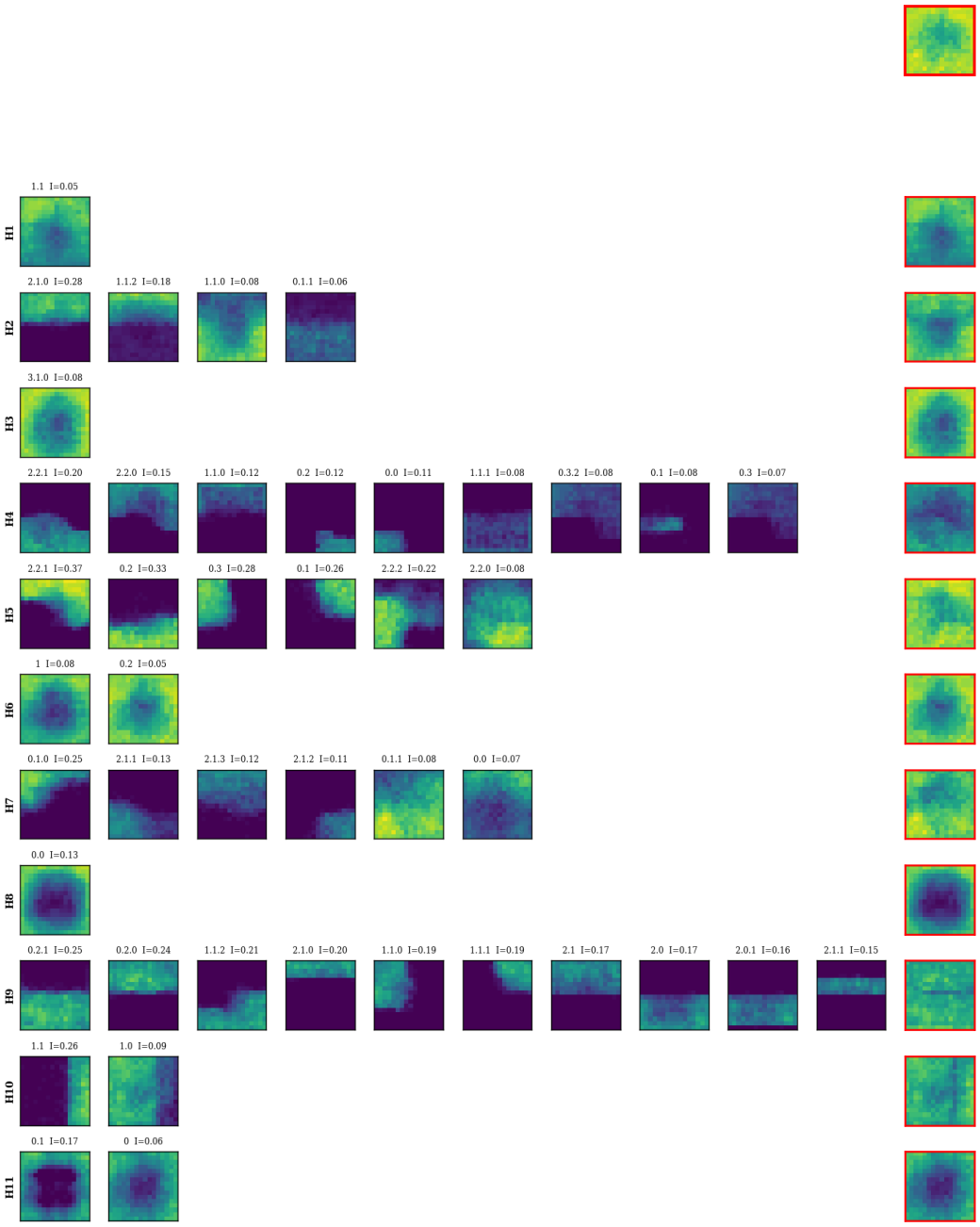


Figure 72: DINOv2 (with registers), $\ell = 0$, Q-side positional leaves.

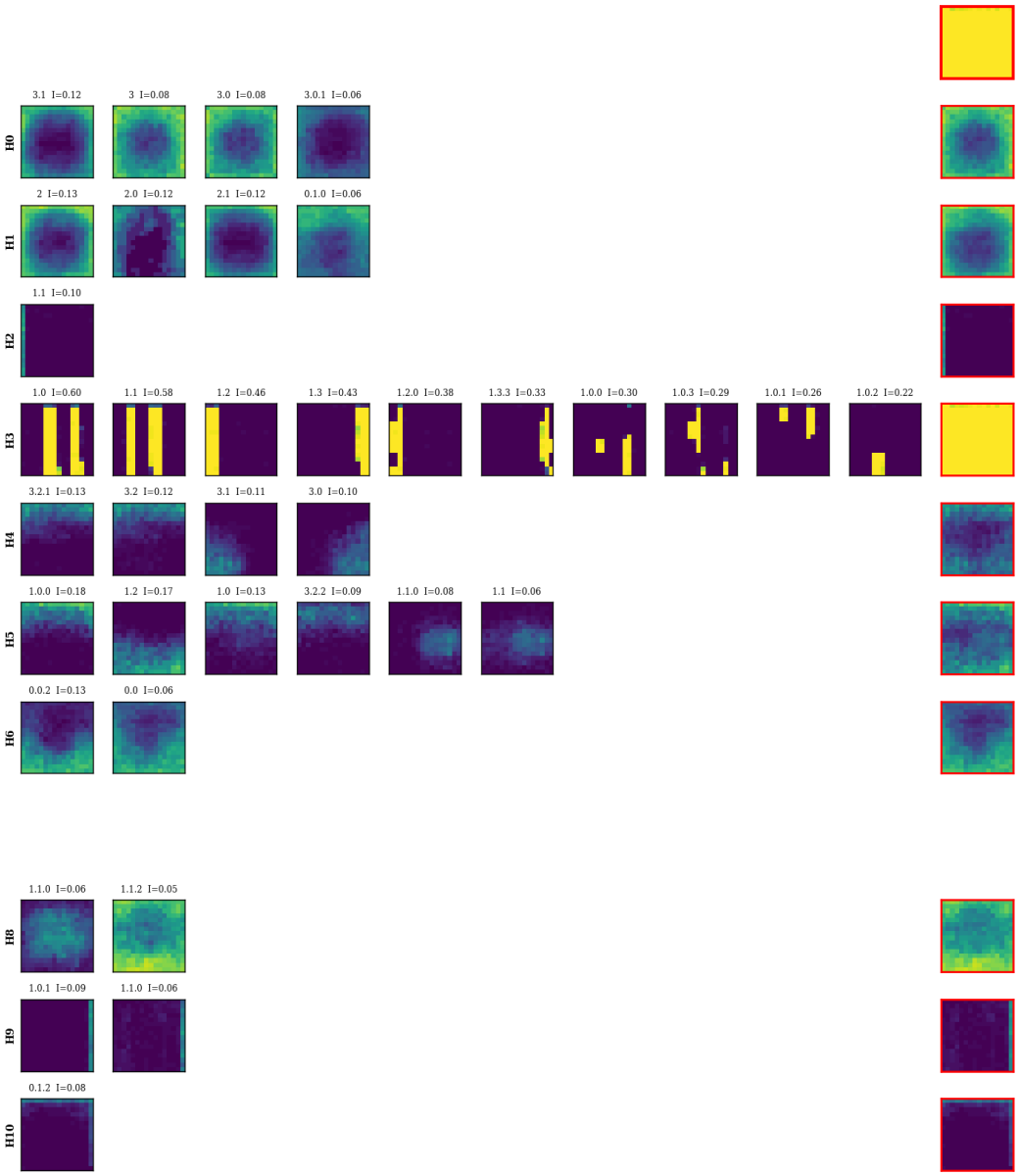


Figure 73: DINOv2 (with registers), $\ell = 2$, Q-side positional leaves.

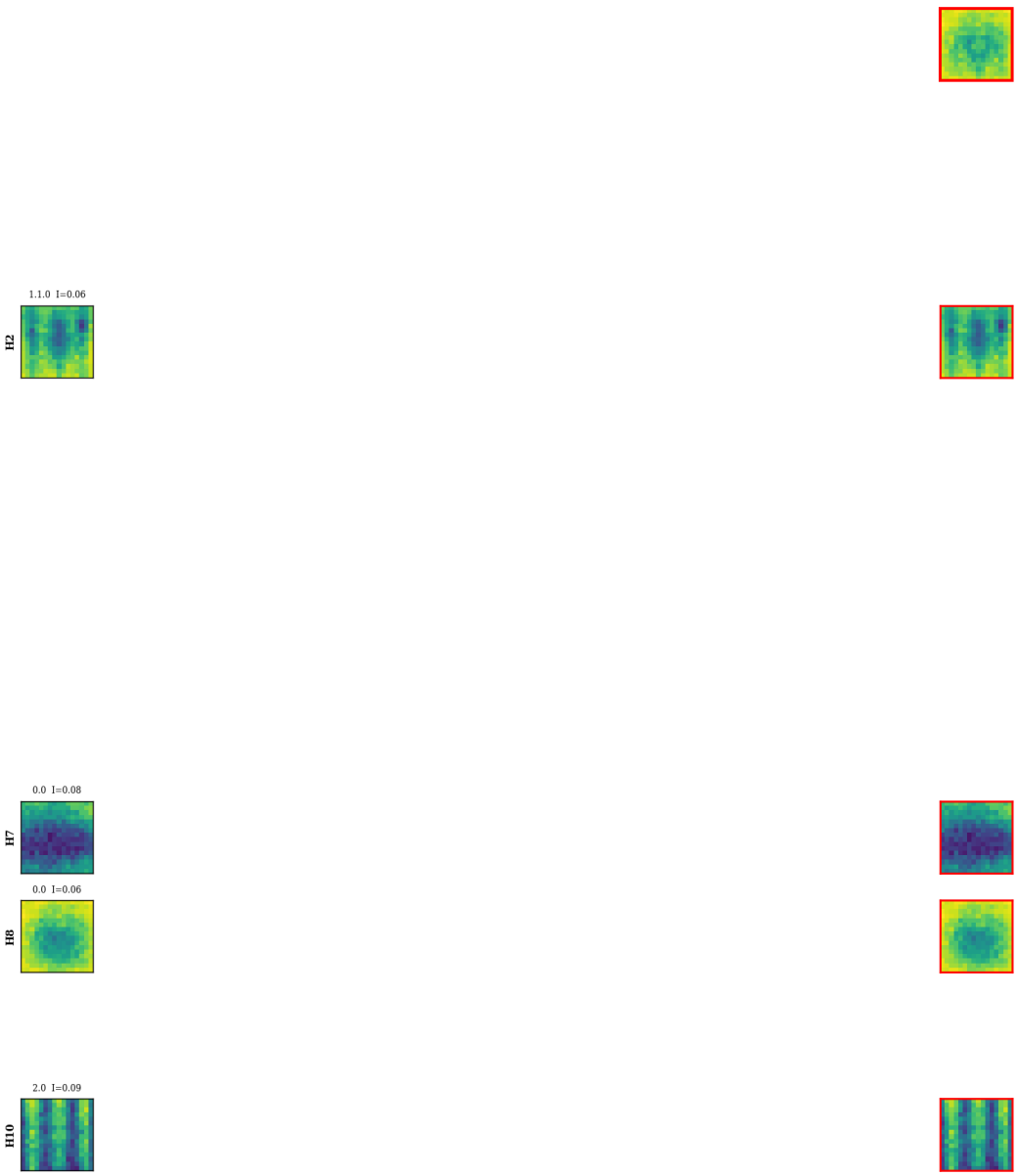


Figure 74: DINOv2 (with registers), $\ell = 0$, K-side positional leaves.

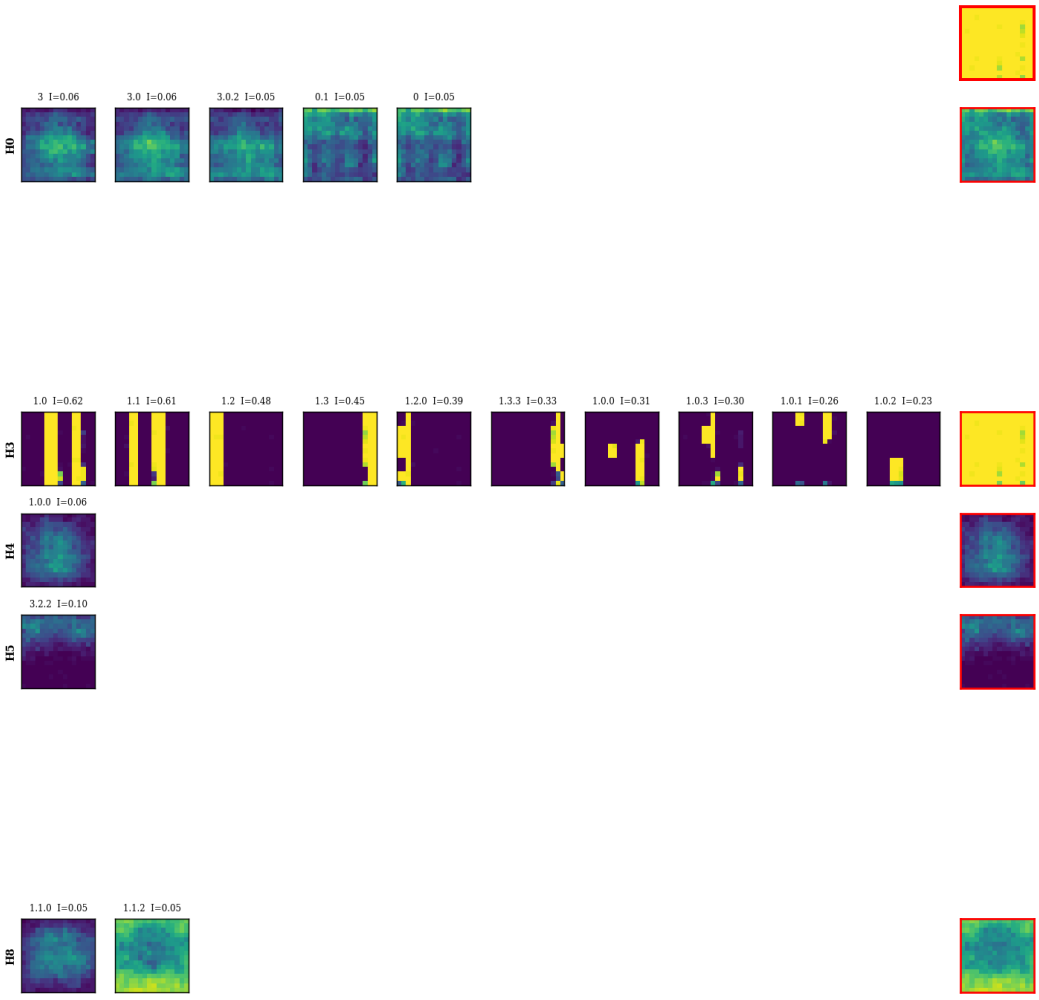


Figure 75: DINOv2 (with registers), $\ell = 2$, K-side positional leaves.

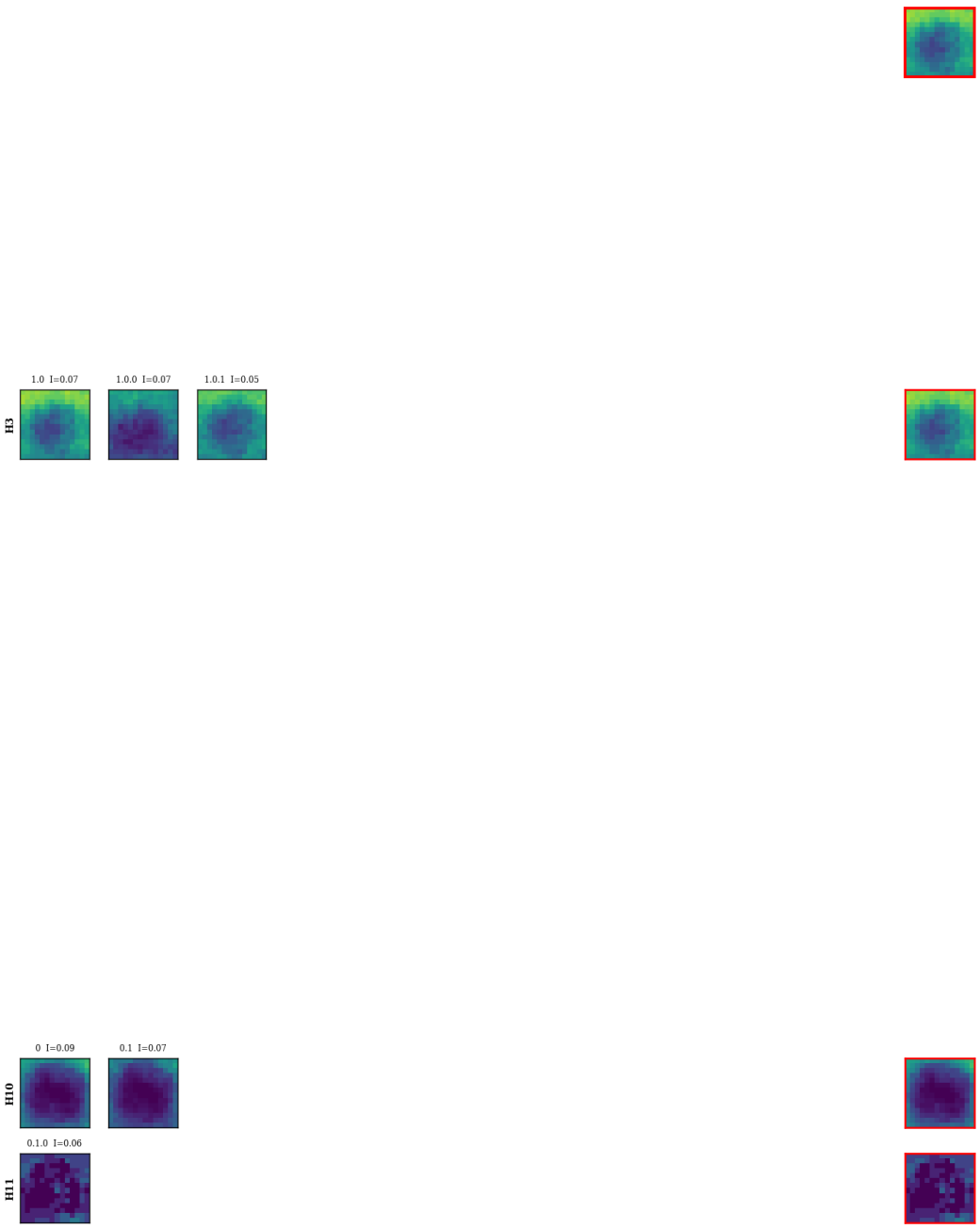


Figure 76: DINOv3, $\ell = 0$, Q-side positional leaves.

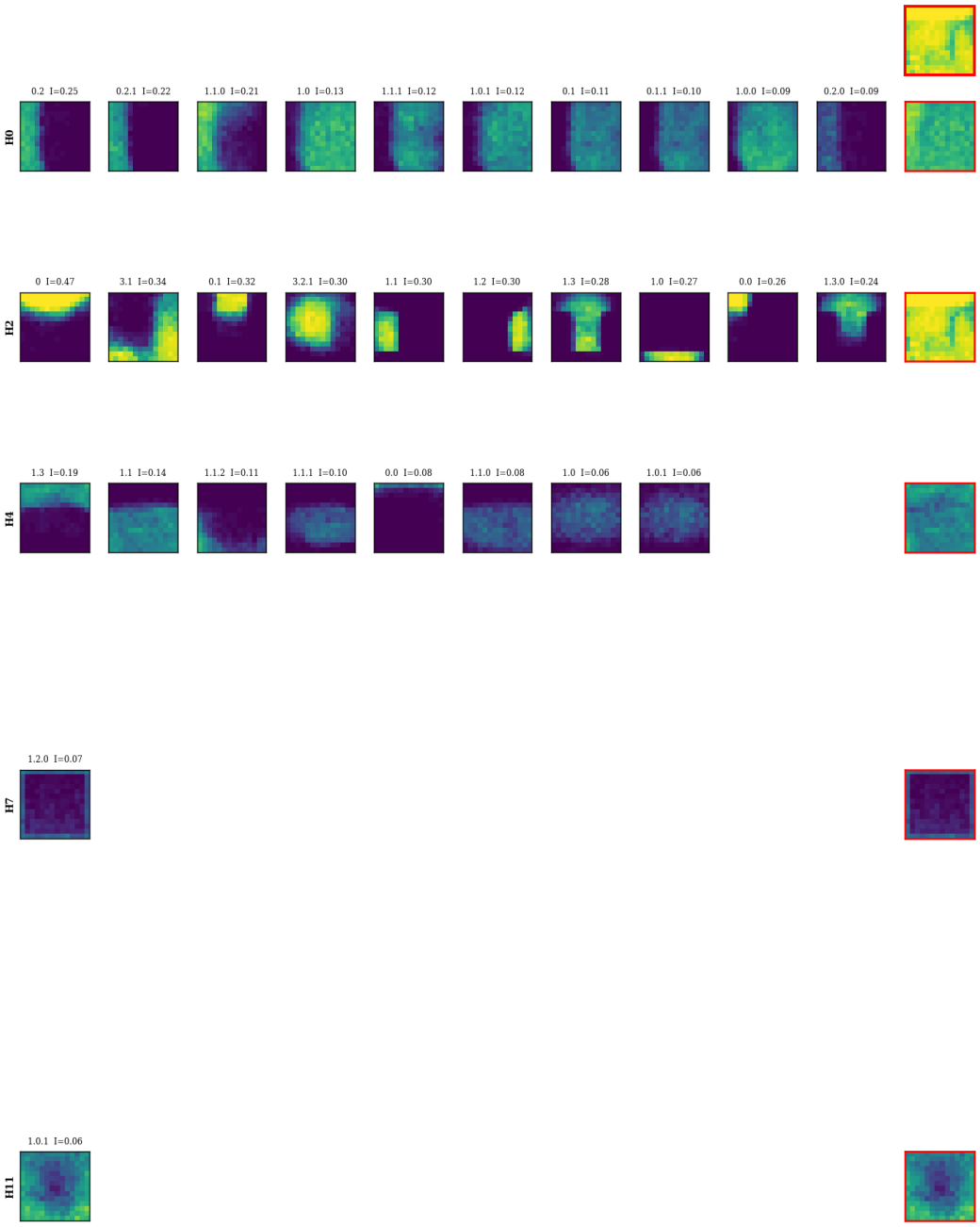


Figure 77: DINOv3, $\ell = 2$, Q-side positional leaves.



Figure 78: DINOv3, $\ell = 0$, K-side positional leaves.



Figure 79: DINOv3, $\ell = 2$, K-side positional leaves.

J Object Localization Details

J.1 Detailed Experimental Setup

This subsection expands the setup in §7 and provides the complete experimental protocol used for Table 2.

Backbones, datasets, and evaluation metric. We evaluate four frozen ViT-B backbones: supervised ViT-B/16 (Dosovitskiy et al., 2021), CLIP ViT-B/16 (Radford et al., 2021), DINOv2+reg ViT-B/14 (Oquab et al., 2024), and DINOv3 ViT-B/16 (Siméoni et al., 2025). Each model is evaluated at its native patch resolution. Following the standard saliency-evaluation protocol, we report mean foreground IoU on ECSSD (Yan et al., 2013), DUT-OMRON (Yang et al., 2013), and DUTS-TE (Wang et al., 2017).

Reference attention baseline. The reference row of Table 2 uses the last-layer `[CLS]`-to-patch attention averaged across heads. Because this baseline involves no calibration, we use the fixed cumulative-attention rule from DINO visualization (Caron et al., 2021), retaining the highest-attention patches whose cumulative mass reaches 60% of the total. We use this rule rather than a fixed threshold on the min-max normalized map, which can be sensitive to the strongly peaked distributions of raw attention maps.

Head and layer baselines. The *head* and *layer* baselines enumerate $6 \times 12 = 72$ and 6 candidate maps, respectively, over layers 6–11. Together with the seven binarization rules of §7, this yields 504 head configurations and 42 layer configurations, each scored by mean IoU over the 500 calibration images. Our node-level selection evaluates between 528 and 945 candidate nodes, depending on the backbone. The map-mean threshold was selected in seven of the eight baseline settings. The only exception was the best-layer baseline for CLIP, which selected the 0.8 cumulative-mass rule.

Evaluation consistency. All methods use the same 224×224 forward pass of the same frozen backbone, the same patch grid, and the same soft-downsampled ground truth. Thus, the comparison does not introduce method-specific differences in input resolution or mask post-processing. No CRF or bilateral solver is applied to any method.

J.2 Qualitative Results

We provide qualitative localization results below (Figures 80–91). Each figure shows, from left to right, the input image, the ground-truth mask, the attention-map reference, the best-head and best-layer baselines, and our single-node method, on six randomly sampled images per dataset.

K Top k Sweep and Inference-Time Trade-off for Object Localization

This section complements §7 with four additional analyses: ① the greedy multi-node selection procedure used for the ablation, ② the full k-sweep across all four backbones and three datasets, ③ a per-k inference-time benchmark, and ④ a comparison between Q-side and K-side aggregation for node selection.

K.1 Greedy Multi-node Selection

The main text selects a single node to isolate the effect of granularity. However, a head may contain multiple object-relevant nodes with complementary localization maps. We therefore additionally consider a simple greedy procedure that selects and aggregates k nodes. Despite its simplicity, this multi-node aggregation improves localization over the single-node setting in many backbone–dataset combinations, suggesting that combining multiple nodes can better exploit object-relevant structure distributed among nodes in the QK tree.



Figure 80: Qualitative localization results on the ECSSD dataset using supervised ViT-B/16.

Directions. Each node $v \in \mathcal{T}^{\ell,h}$ is associated with two *directions*: an *object* direction given by $h^{\mathrm{Q}}_{I,v}$ and a *background* direction given by its complement $1 - h^{\mathrm{Q}}_{I,v}$. This allows both foreground- and background-selective nodes to contribute to localization. We allow each node to contribute at most one direction. Including both directions of the same node would produce a trivial cancellation effect, since their sum is constant and only shifts the effective threshold.

Greedy selection. Starting from an empty set, we iteratively add the (node, direction) pair that maximizes the mean IoU of the averaged map over the same 500 DUTS-TR calibration images used in §7. At each step, the averaged map is thresholded at 0.5. After selecting k pairs, the resulting set is fixed and reused at test time. For an unseen image, we average the k selected maps and threshold the result at 0.5. Note that no additional post-processing, such as CRF or bilateral filtering, is applied.

K.2 Full k-sweep

We sweep $k \in \{1, 2, 3, 5, 10, 15, 20\}$ for all four backbones. Tables 9–11 report the results for each dataset, with rows corresponding to backbones and columns to values of k.

The main text uses a single node ($k = 1$). Adding more nodes improves localization in many backbone–dataset combinations, although the gains generally diminish as k increases. The largest

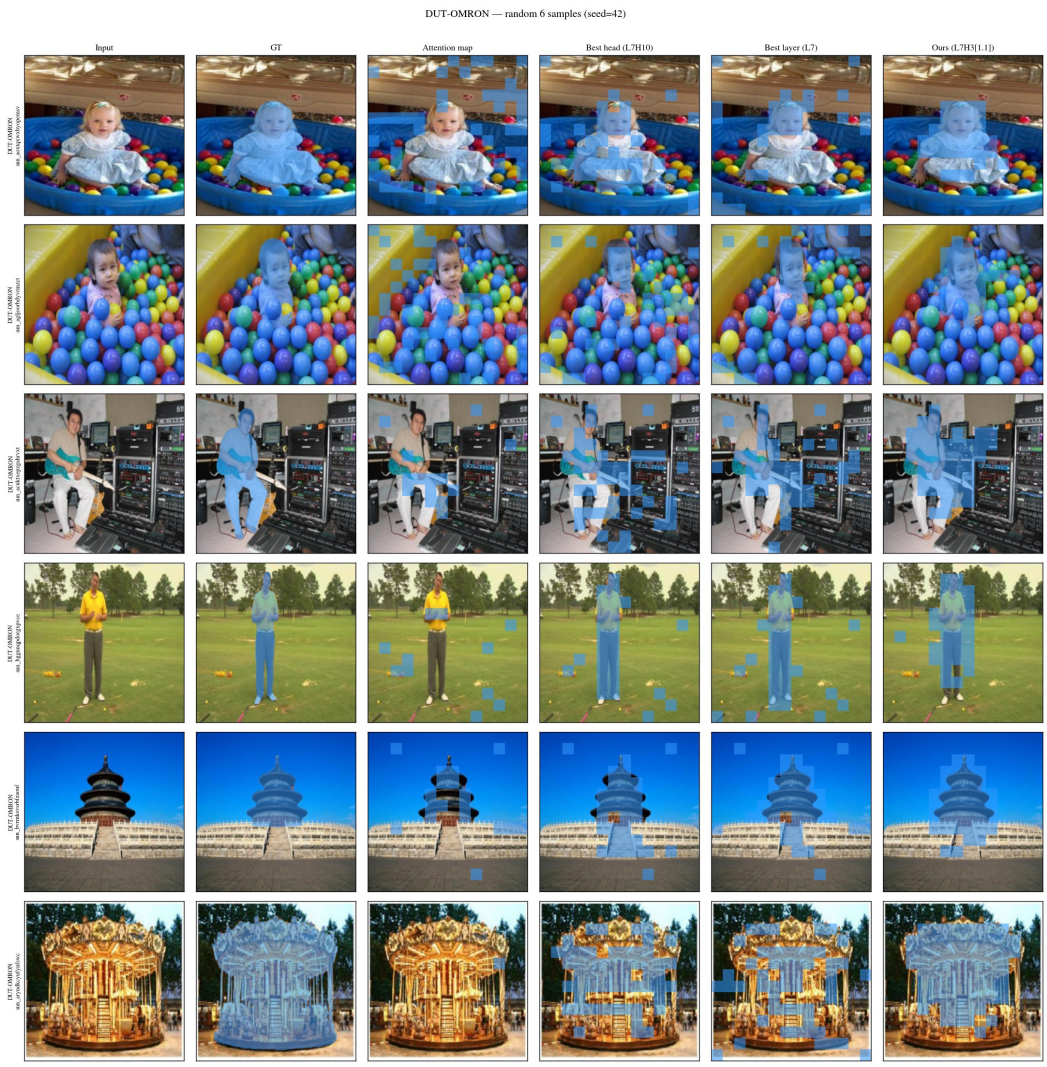


Figure 81: Qualitative localization results on the DUT-OMRON dataset using supervised ViT-B/16.

Table 9: ECSSD mean foreground IoU at varying k.

Backbone	k=1	k=2	k=3	k=5	k=10	k=15	k=20
ViT-B/16 (sup.)	.469	.535	.512	.558	.627	.612	.640
CLIP B/16	.569	.572	.583	.590	.615	.624	.626
DINOv2+reg B/14	.647	.663	.655	.650	.634	.631	.653
DINOv3 B/16	.682	.685	.686	.693	.704	.704	.709

improvements occur for supervised ViT, where the increase over $k = 1$ reaches approximately .10–.17 IoU. For CLIP, DINOv2+reg, and DINOv3, the gains are generally smaller. DUT-OMRON shows less consistent improvement with increasing k than ECSSD and DUTS-TE. Overall, multi-node aggregation can improve upon the single-node setting, and most of the improvement is already obtained by $k = 10$.

K.3 Per-k Inference-time Benchmark

At test time, the computational cost depends largely on the number of *distinct heads* that must be evaluated for each image, that is, the number of heads containing at least one selected node.

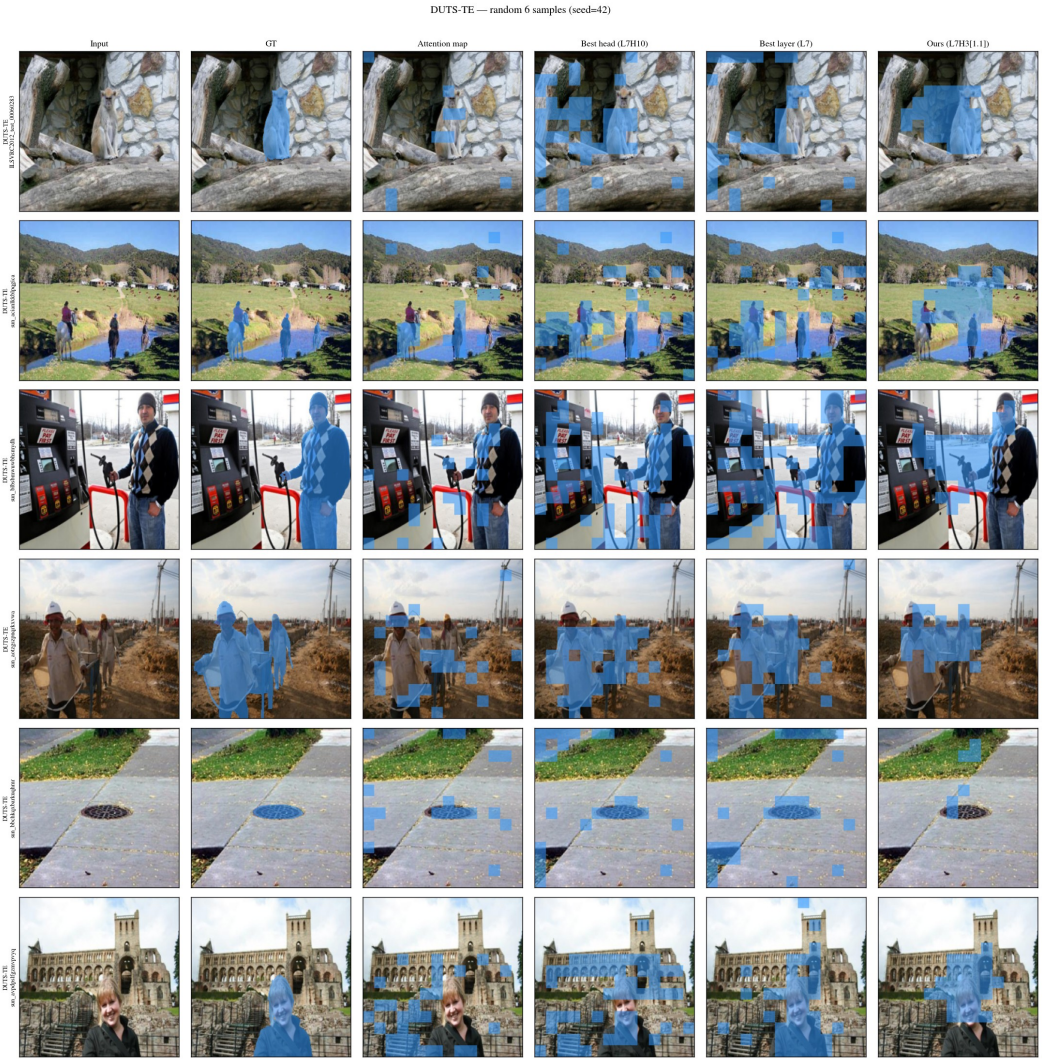


Figure 82: Qualitative localization results on the DUTS-TE dataset using supervised ViT-B/16.

Table 10: DUT-OMRON mean foreground IoU at varying k.

Backbone	k=1	k=2	k=3	k=5	k=10	k=15	k=20
ViT-B/16 (sup.)	.336	.344	.347	.376	.432	.431	.439
CLIP B/16	.400	.381	.381	.355	.359	.385	.387
DINOv2+reg B/14	.448	.454	.440	.432	.410	.416	.428
DINOv3 B/16	.449	.469	.443	.454	.485	.478	.485

Multiple selected nodes from the same head can share the same backbone forward pass and head-level computation, including the random-forest traversal over the $T \times T$ query-key pairs. Table 12 reports wall-clock inference times measured on 50 DUTS-TE images per backbone using a single H100 GPU.

The per-image inference time generally increases with the number of distinct heads, while the number of distinct heads grows more slowly than k because greedy selection can choose multiple nodes from the same head. The single-node setting used in the main text evaluates only one head and has the lowest measured inference time. At $k = 10$, inference requires approximately 2.1–2.7 s per image

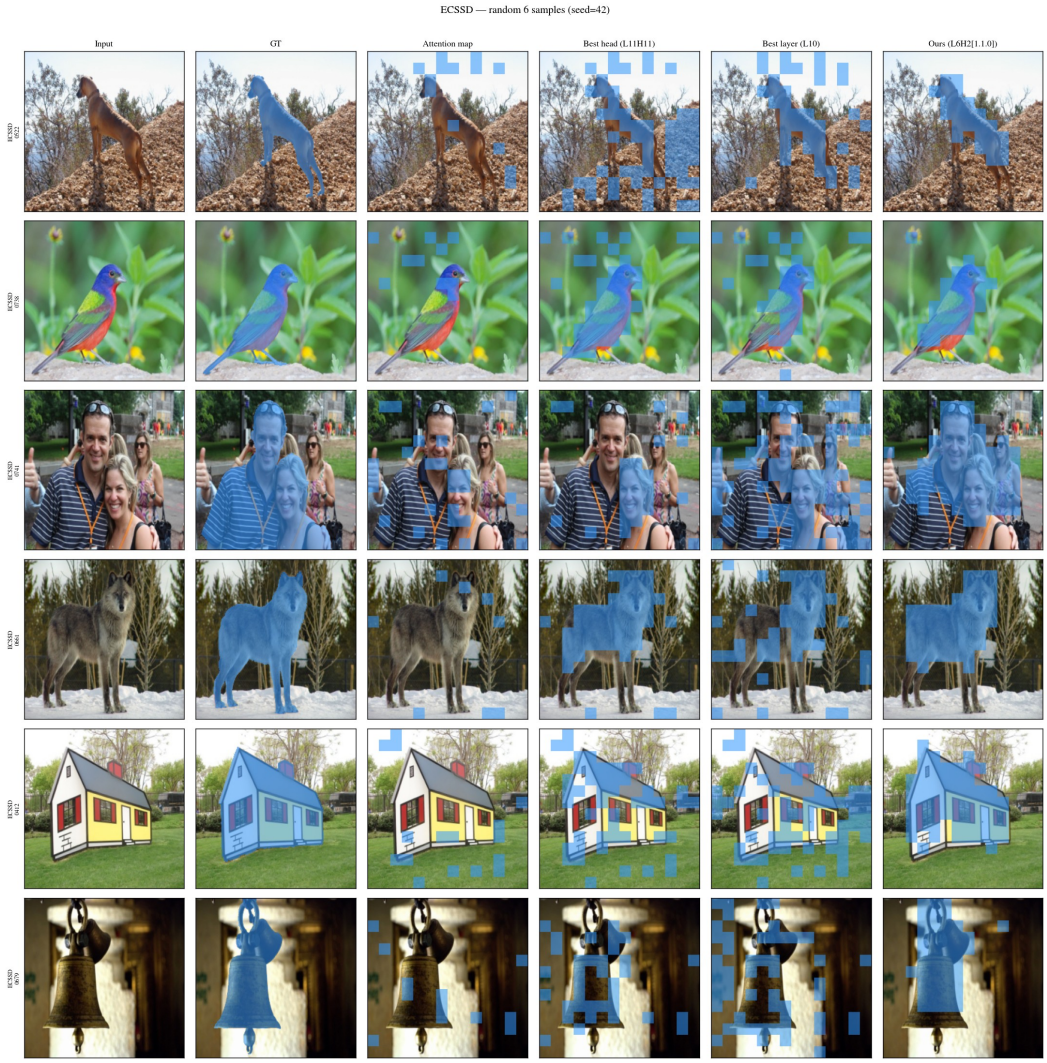


Figure 83: Qualitative localization results on the ECSSD dataset using CLIP ViT-B/16.

Table 11: DUTS-TE mean foreground IoU at varying k.

Backbone	k=1	k=2	k=3	k=5	k=10	k=15	k=20
ViT-B/16 (sup.)	.349	.376	.410	.432	.471	.485	.478
CLIP B/16	.435	.447	.455	.468	.491	.499	.498
DINOv2+reg B/14	.481	.518	.520	.518	.514	.517	.513
DINOv3 B/16	.521	.553	.573	.575	.575	.570	.572

across the four backbones. Beyond this range, additional selected nodes increase computation without yielding consistent IoU improvements across datasets.

K.4 Q-side vs. K-side Aggregation

The main experiments use the Q-side heatmap $h^{\mathrm{Q}}_{I,v}$ for node selection. To assess the effect of this choice, we repeat the complete node-selection pipeline of §7 using $h^{\mathrm{K}}_{I,v}$ instead, including calibration, ranking, and inference. The K-side map is the column-attention-weighted counterpart of the Q-side map defined in Appendix C.5. All remaining settings are unchanged, including the 500-image DUTS-TR calibration set, threshold 0.5, and the single-node setting $k = 1$.

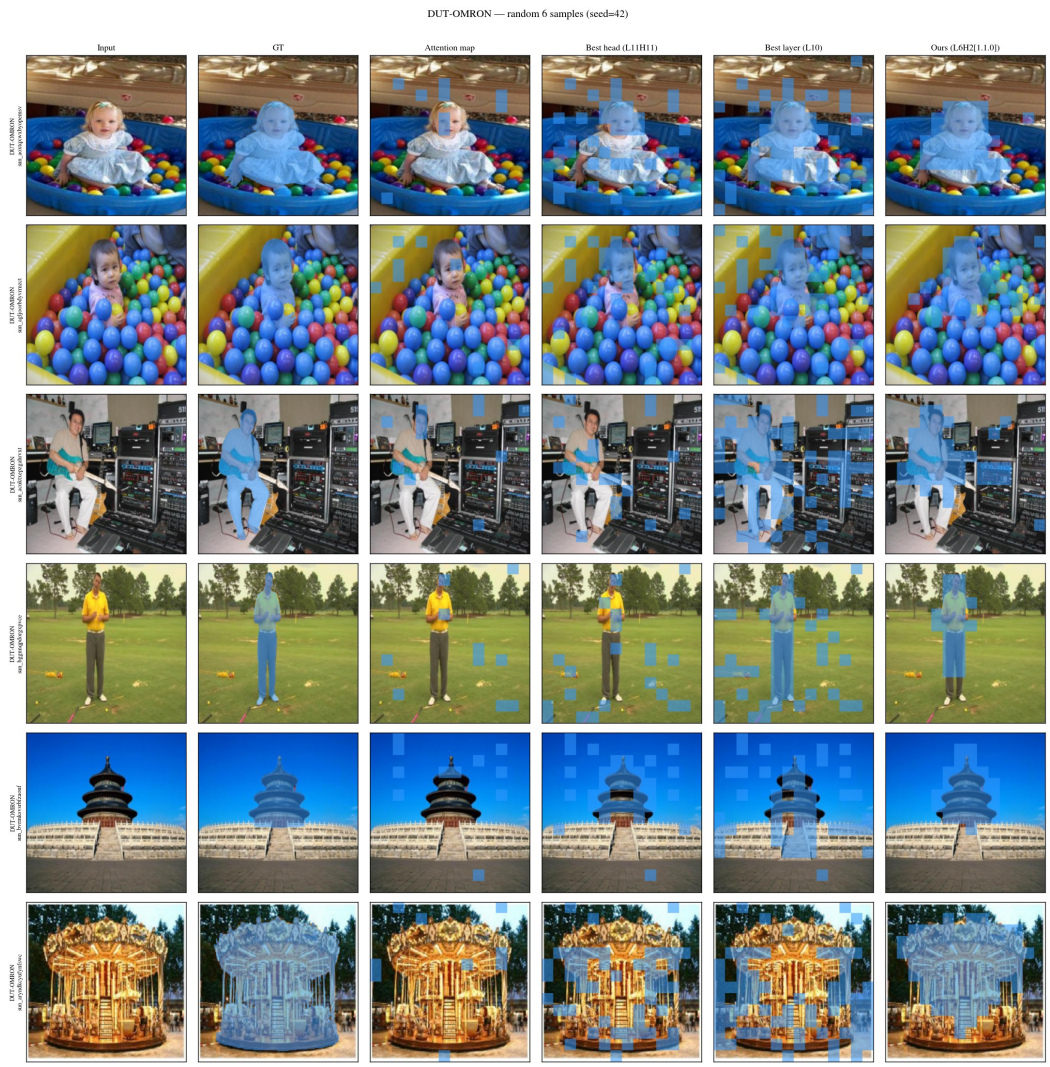


Figure 84: Qualitative localization results on the DUT-OMRON dataset using CLIP ViT-B/16.

Table 12: Per-image inference time (ms) and number of distinct heads at varying k.

Backbone	Metric	k=1	k=2	k=3	k=5	k=10	k=15	k=20
ViT-B/16 (sup.)	#heads	1	2	2	4	9	12	16
	ms	332	547	553	1240	2248	2709	3383
CLIP B/16	#heads	1	2	3	5	10	13	17
	ms	261	558	821	1335	2663	3485	4628
DINOv2+reg B/14	#heads	1	2	3	3	7	8	12
	ms	482	762	1127	1036	2185	2364	3818
DINOv3 B/16	#heads	1	2	3	5	8	9	12
	ms	323	645	920	1517	2238	2420	2982

Table 13 compares the K-side results with the Q-side values from Table 2 for supervised ViT, DINOv2+reg, and CLIP. The two aggregation schemes yield similar results overall. The Q-side map performs better in eight of the nine settings, with margins of at most .009 for supervised ViT and .025 for CLIP, and up to .036 for DINOv2+reg. The qualitative conclusion of §7 is unchanged. For

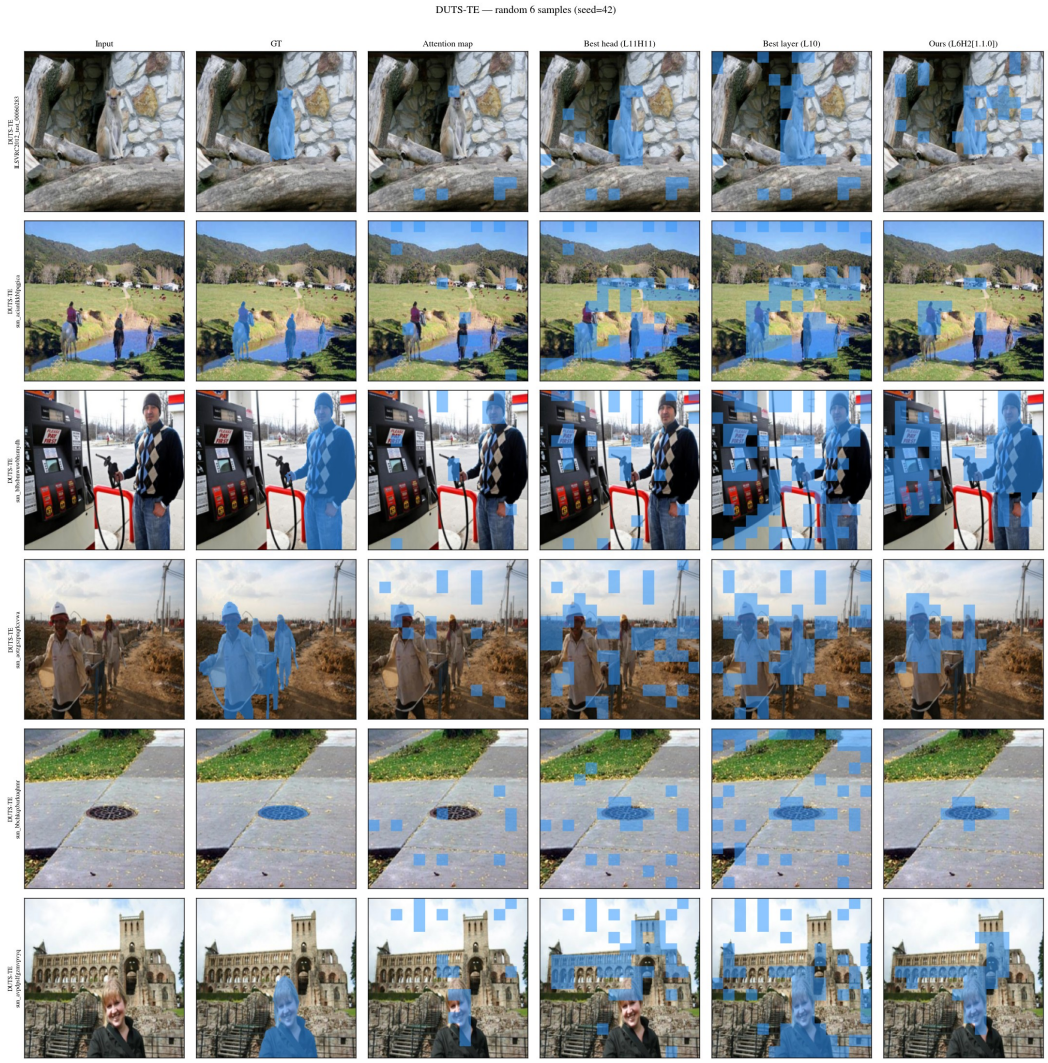


Figure 85: Qualitative localization results on the DUTS-TE dataset using CLIP ViT-B/16.

Table 13: Mean foreground IoU when the node-selection pipeline of §7 is repeated with the K-side heatmap $h^{\mathrm{K}}_{I,v}$ instead of the default Q-side map $h^{\mathrm{Q}}_{I,v}$. Δ denotes K-side minus Q-side.

Backbone	Dataset	Q-side (h^{Q})	K-side (h^{K})	Δ
ViT-B/16 (sup.)	ECSSD	.469	.462	$-.007$
	DUT-OMRON	.336	.327	$-.009$
	DUTS-TE	.349	.357	$+.008$
DINOv2+reg B/14	ECSSD	.647	.620	$-.027$
	DUT-OMRON	.448	.419	$-.029$
	DUTS-TE	.481	.445	$-.036$
CLIP B/16	ECSSD	.569	.555	$-.014$
	DUT-OMRON	.400	.385	$-.015$
	DUTS-TE	.435	.410	$-.025$

these backbones, a single node selected from either side still outperforms the raw attention-map and best-head baselines.



Figure 86: Qualitative localization results on the ECSSD dataset using DINOv2+reg B/14.



Figure 87: Qualitative localization results on the DUT-OMRON dataset using DINOv2+reg B/14.

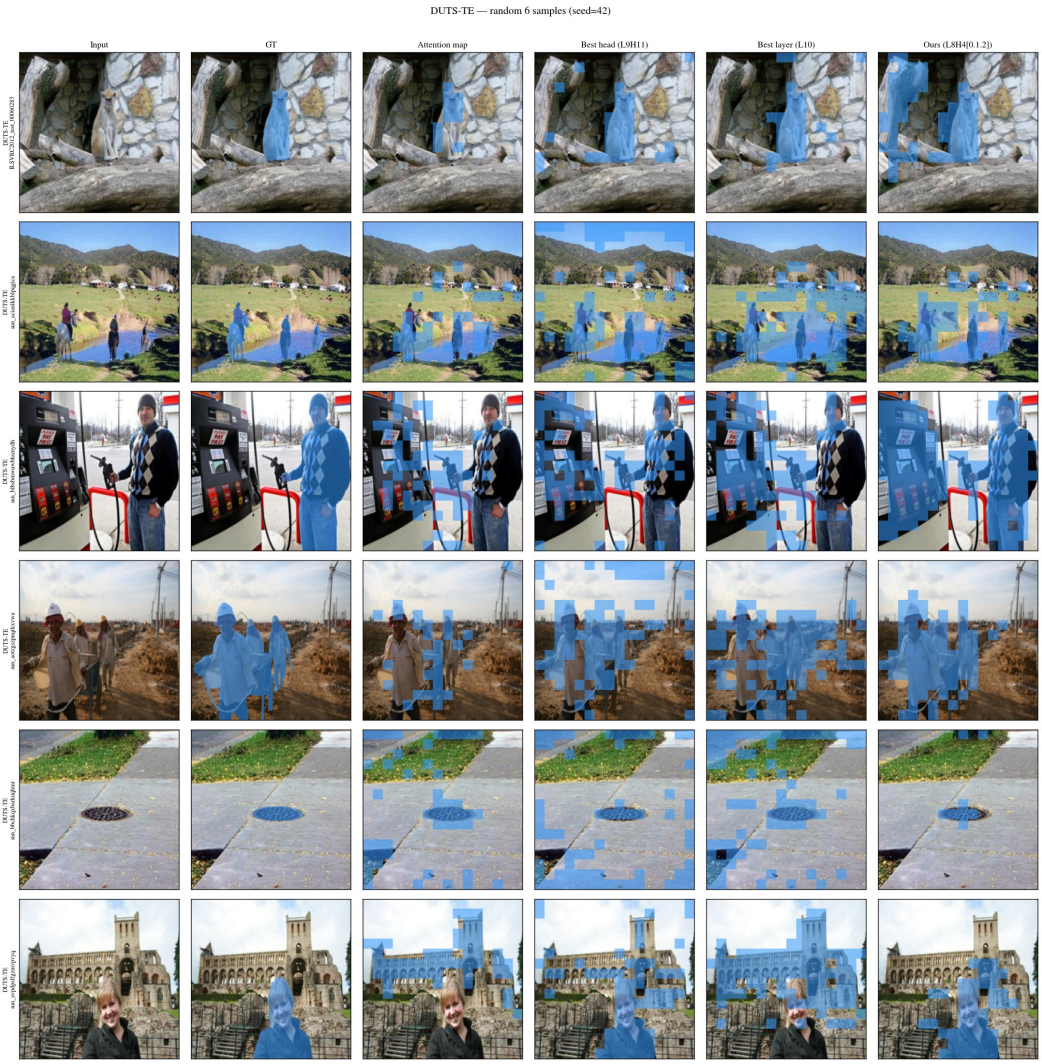


Figure 88: Qualitative localization results on the DUTS-TE dataset using DINOv2+reg B/14.

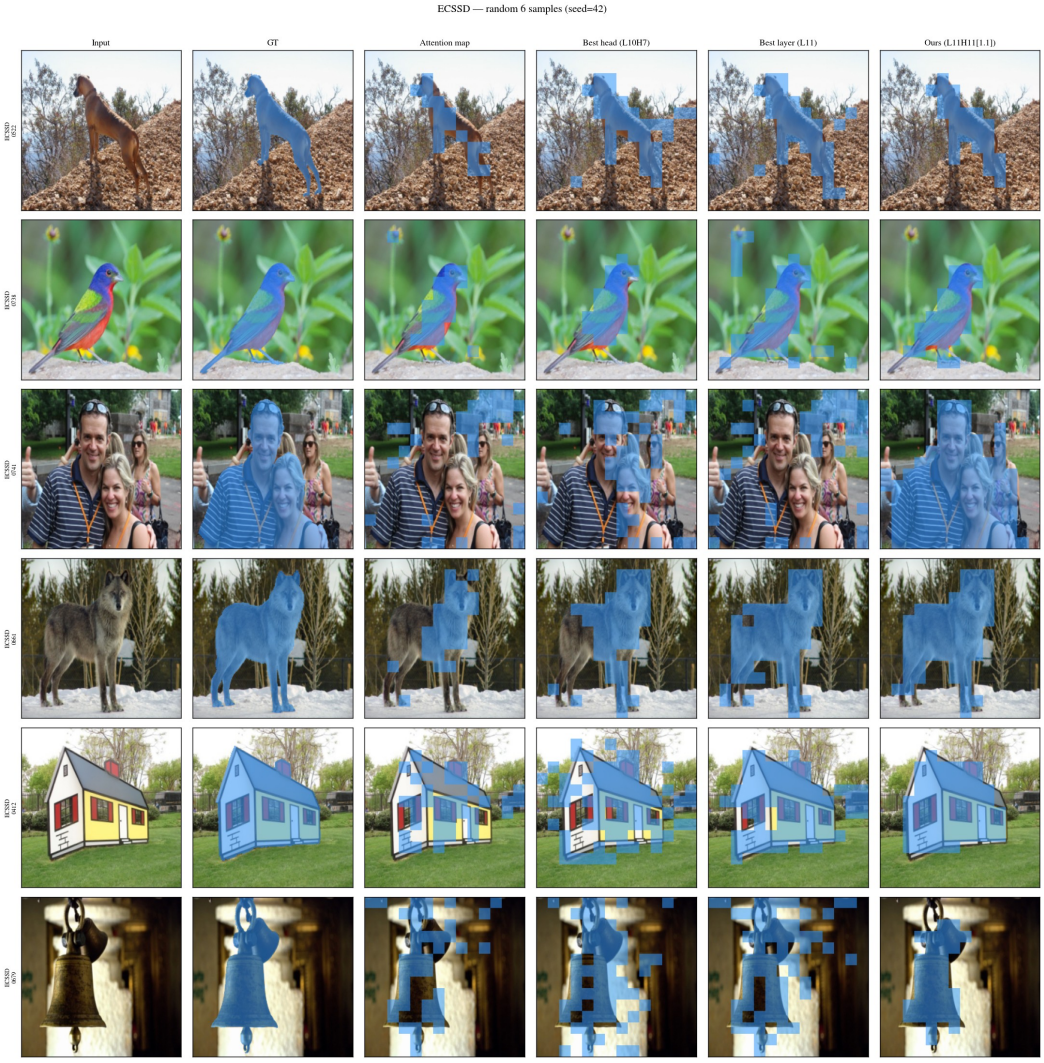


Figure 89: Qualitative localization results on the ECSSD dataset using DINOv3 B/16.

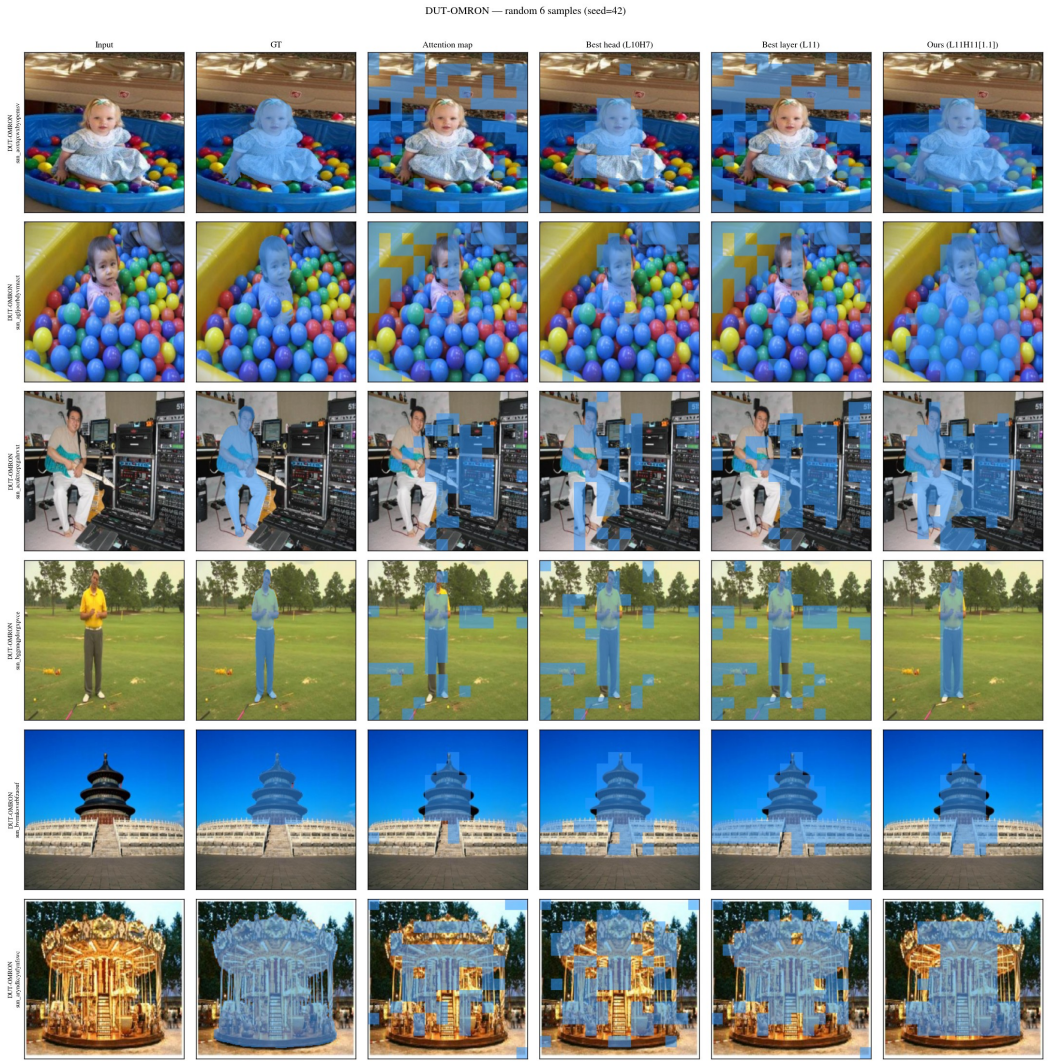


Figure 90: Qualitative localization results on the DUT-OMRON dataset using DINOv3 B/16.

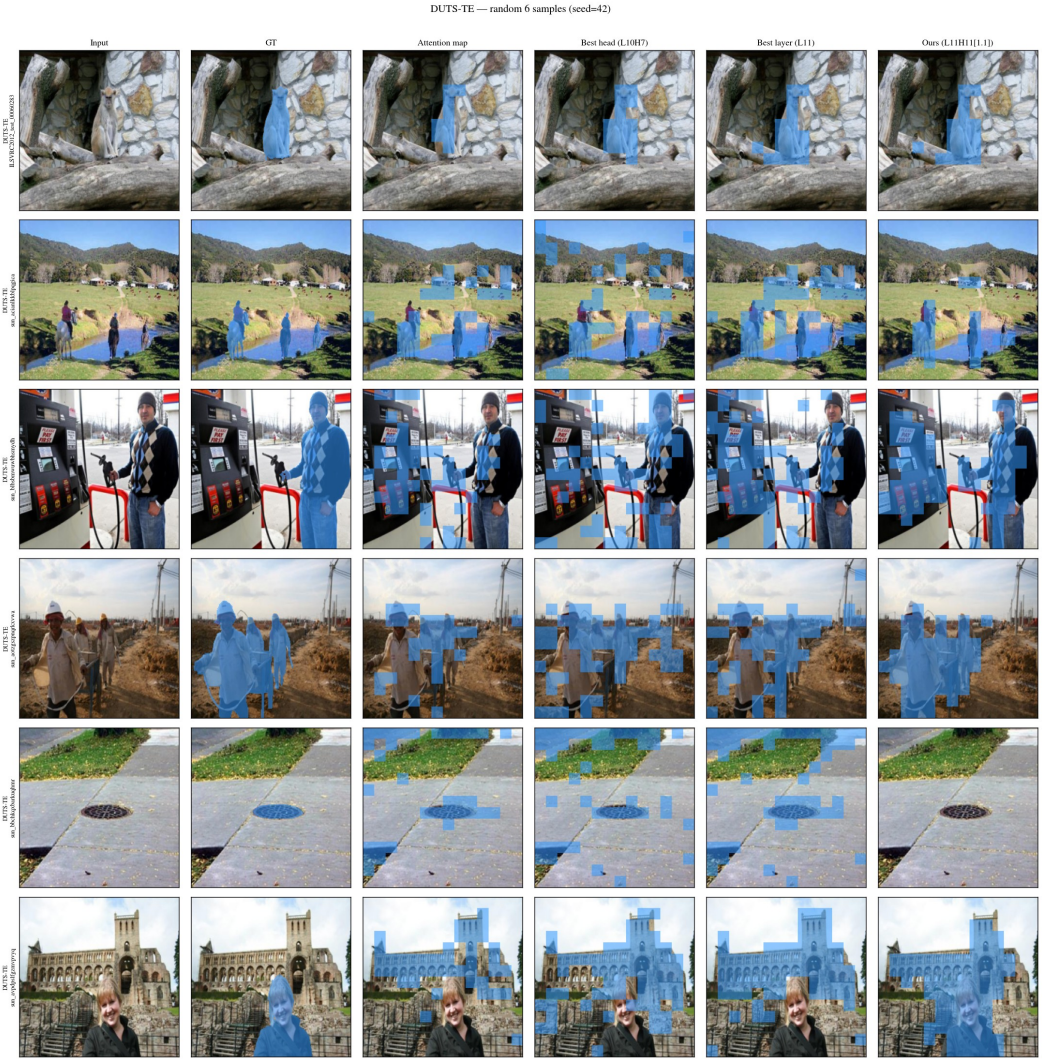


Figure 91: Qualitative localization results on the DUTS-TE dataset using DINOv3 B/16.