OBJECTS AS ADDRESSES: INSTANCE BINDING IN VISION TRANSFORMER ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans naturally parse visual scenes into coherent individual objects rather than scattered features, a capability cognitive science calls object binding. While recent work shows that pretrained Vision Transformers (ViTs) intrinsically acquire this ability, how native self-attention computations mechanistically keep object instances separate remains unresolved. By studying this across eight pretrained ViTs, we found that instance separation is expressed more strongly in attention routing than in token content: selected query–key relations distinguish individual objects more reliably than value similarity or residual representations. Inside heads, patches belonging to one individual tend to read from a common image region, which we term their *attention address*. Visually distinct parts of one person share an address, whereas identical duplicated instances receive separate, movable addresses. These shared destinations arise in part because query interactions with the spatial structure of keys direct attention toward consistent regions for one object and elsewhere for another. In DINOv2 and most other backbones we tested, editing this query component at the selected head redirects its attention address toward another individual. Grouping patches by this query component separates same-category instances, reaching .615 FG-ARI versus .289 for the same clustering of final-layer features, which tend to merge them.

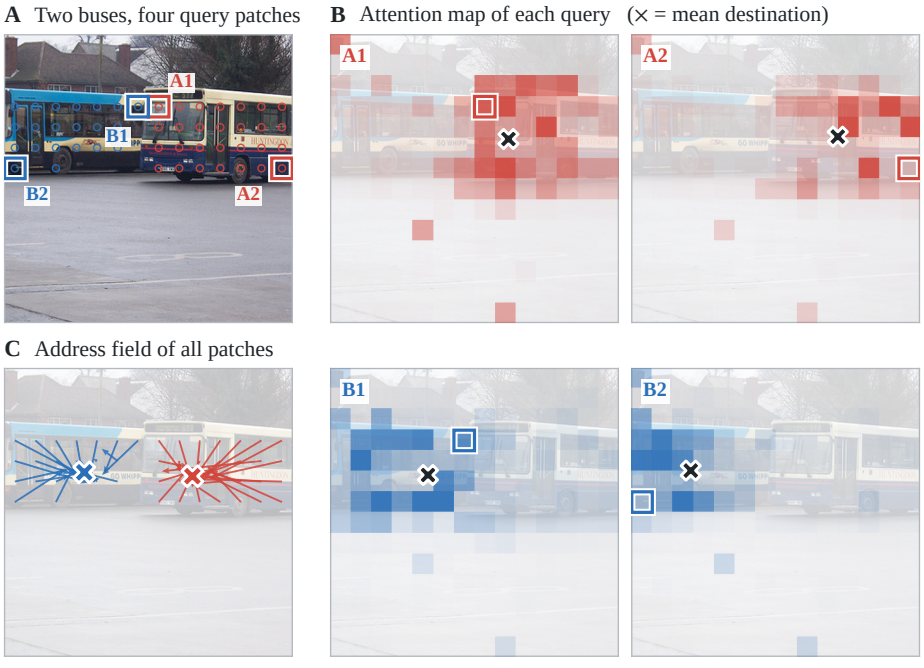

Figure 1: **Objects as addresses.** The cross marks the mean destination of each map. Colors use ground truth only to show which patches belong together. (A) Four patches query the same frozen attention head, two on each bus. (B) Patches on one bus can look different, yet their complete attention maps read from a common region. (C) Drawing an arrow from every bus patch to its address turns the head into an address field, in which the patches of each bus point to one region.

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.

1 INTRODUCTION

An image can contain two zebras. However, labeling every relevant patch as *zebra* is not enough: the model must still associate each head, torso, and leg with the correct individual. This is the visual binding problem, representing not only *what* features are present, but *which features belong together* (Treisman & Gelade, 1980; Greff et al., 2020). This problem was tackled by Li et al. (2025), who recovered patch-level same-object membership, ISSAMEOBJECT, with a learned quadratic probe and connected it to attention. In contrast, our aim is to identify how existing ViT attention distinguishes individuals, then to explain and manipulate the part of its computation that directs where their patches read. We focus on objects of the same category. Nearby patches often belong to the same object, as do patches with similar semantics, and either cue can obscure whether an attention head distinguishes individuals. We therefore compare same-instance pairs of patches with pairs from different instances of one category in the same image and at exactly the same relative offset (same distance and direction). We call this controlled problem *object individuation*.

We used Pascal VOC for development and held-out evaluation, and COCO for transfer and later analyses. Labels define the test and are used to select the layers, heads, and relations on the VOC development split. Evaluation uses disjoint held-out VOC images. COCO is not used for component selection unless stated otherwise. No binding predictor is trained.

By analyzing the results, we identified heads whose reading distinguishes individuals better than residual or value similarity. Following their attention reveals a common destination for patches of one object. We then examined how a query interaction with spatial structure in the keys helps produce that destination. Our contributions are:

1. **A controlled test that localizes instance-sensitive attention.** Across eight pretrained ViTs, query/key comparisons and attention separate individuals more strongly than value or residual similarity. This advantage concentrates in particular heads and layers.
2. **A shared reading destination for parts of one object.** Patches of an individual tend to read from a common region, their *attention address*. Identical copies receive different addresses, and an address moves with its object.
3. **An interpretable spatial request within attention.** A query interaction with the keys' spatial structure specifies where it would read, its *spatial request*. This request retains instance discrimination across several backbones. In DINOv2, its destination predicts the attention address, and editing it redirects the head toward another individual.

2 RELATED WORK

Binding and object representations. The binding problem asks how visual features are organized into coherent objects, rather than represented independently (Treisman & Gelade, 1980; Pylyshyn, 1989; Kahneman et al., 1992; Greff et al., 2020). Li et al. (2025) decoded same-object membership from frozen ViTs with a quadratic probe and linked this signal to attention. Huang et al. (2026) formalized binding information, that is, which features belong to the same object, and measured it in ViT representations, while Tartaglini & A Lepori (2026) found mostly early heads that track continuity in synthetic stimuli and contribute to binding. We ask a more specific question: which native attention computations keep individuals of the same category apart, and how?

Object-centric learning and pretrained ViTs. Object-centric learning makes object assignment explicit through representations such as slots (Locatello et al., 2020; Seitzer et al., 2023). Yet instance identity remains challenging: SemanticSlots (Sabri et al., 2026) identified ambiguity between identical entities as a limitation of semantic retrieval, while SlotContrast (Manasyan et al., 2025) used temporal consistency to preserve object identity across frames. Decoder comparisons in DINOSAUR-style systems (Seitzer et al., 2023) also showed that the decoding choice can substantially change instance grouping. This motivates asking what instance information is already present in the pretrained ViT before an object-centric module decides what to preserve. Self-supervised ViTs already expose object structure through attention and feature affinities (Caron et al., 2021; Siméoni et al., 2021; Wang et al., 2022; Melas-Kyriazi et al., 2022), while Object-DINO (Rawlekar et al., 2026) finds complementary object information across Q, K, and V.

Attention, space, and binding. Zhang et al. (2024) found that Q/K carries strong spatial structure while V is more semantic. Pan et al. (2024) decomposed the native Q/K interaction and found that early layers attend more to similar tokens and late layers to dissimilar, contextual ones. Prior work also studied spatial structure in attention (Cordonnier et al., 2020; d'Ascoli et al., 2021; Kim et al., 2023). We examine how a pretrained head uses that structure to distinguish instances. Related mechanisms appear in language and VLMs: binding IDs link entities to attributes (Feng & Steinhardt, 2024), while spatial IDs let VLMs retrieve particular visual objects (Assouel et al., 2026) and reason about their locations (Kang et al., 2026). We study this inside the frozen visual encoder itself: which native attention components distinguish individual objects, where those objects are read from, and whether editing the query component that sets this reading redirects the head to another instance.

3 Measuring object individuation in native attention

We first compare the different parts of attention to determine where the same-instance signal appears.

3.1 Where to read and what to carry

Li et al. (2025) predict whether two patches h_i and h_j belong to the same object with a learned bilinear score (probe) $h_i^\top W h_j$, where W is a probe matrix learned from same-object labels. However, a ViT already compares patches through a related score. In each attention head (Vaswani et al., 2017), the query of a patch is compared with the keys of all patches to decide where it reads, and a weighted average of their values is what it retrieves. Elhage et al. (2021) described these two roles as a query–key circuit, which computes the attention pattern, and an output–value circuit, which decides what is moved. For one attention head, let h_i and h_j be the normalized input representations of patches i and j, let q, k, and v denote queries, keys, and values, and let W_Q, W_K, and W_V be their projection matrices. Let d be the query–key dimension. Suppressing affine biases,

$$\begin{aligned} q_i = W_Q h_i, \quad k_i = W_K h_i, \quad v_i = W_V h_i, \\ q_i^\top k_j = h_i^\top W_Q^\top W_K h_j, \\ A_{ij} = \mathrm{softmax}_j(q_i^\top k_j / \sqrt{d}), \end{aligned} \tag{1}$$

where the softmax normalizes over destinations j, and the head returns $o_i = \sum_j A_{ij} v_j$. The probe of Li et al. (2025) learns W from object labels. The head already has its own learned comparison, $W_Q^\top W_K$, which we can test directly. Similarity between two queries (Q–Q) or two keys (K–K) offers another way to test the same head. We therefore ask: which representations and operations of the frozen head expose the same-instance relation?

To answer this, we compared patches through their residual representations (H–H), queries (Q–Q), keys (K–K), and values (V–V), using cosine similarity. We also compared the query–key scores ($Q \leftrightarrow K$) and attention weights of the head, averaging both directions because same-instance membership is symmetric. Appendix A gives the scoring details. We call query similarity, key similarity, the query–key relation, and attention the *routing-side* (Q/K) relations.

3.2 A controlled test of object individuation

We used the following experimental setup. A positive pair contains two patches from one annotated instance, while a negative pair contains patches from two different instances of the same category. All patches belong to foreground objects in the same image, and positive and negative pairs occur equally often at every relative offset, including its direction. For example, if two patches of zebra A are separated by $(+3, +1)$, the comparison includes equally many cross-instance pairs with that offset. By doing this, category agreement, foreground membership, distance, and direction cannot solve the task. Each relation scores every pair, and we report the AUC of ranking positive above negative pairs, per image. Without offset matching, proximity alone reaches .79 AUC on same-category COCO pairs, and matching reduces it to chance. Figure 2 illustrates the cues our controlled test removes.

We evaluated frozen ViT-B backbones trained with DINO, DINOv2, DINOv2 with registers, DINOv3, MAE, CLIP, SigLIP, and supervised ImageNet classification. We split VOC into a development set, used to select components, and a disjoint held-out set used for evaluation. For each relation,

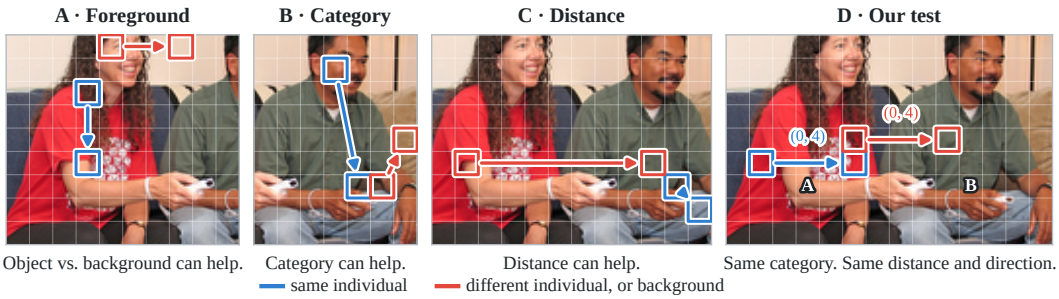

Figure 2: **Testing which individual, beyond easier cues.** In an unrestricted patch-pair task, object–background differences (A), category differences (B), or unequal distances (C) can help distinguish pair labels. Our test compares patches on objects of the same category and balances same- and different-instance pairs at every offset, such as the $(0, +4)$ shared by the two arrows in D.

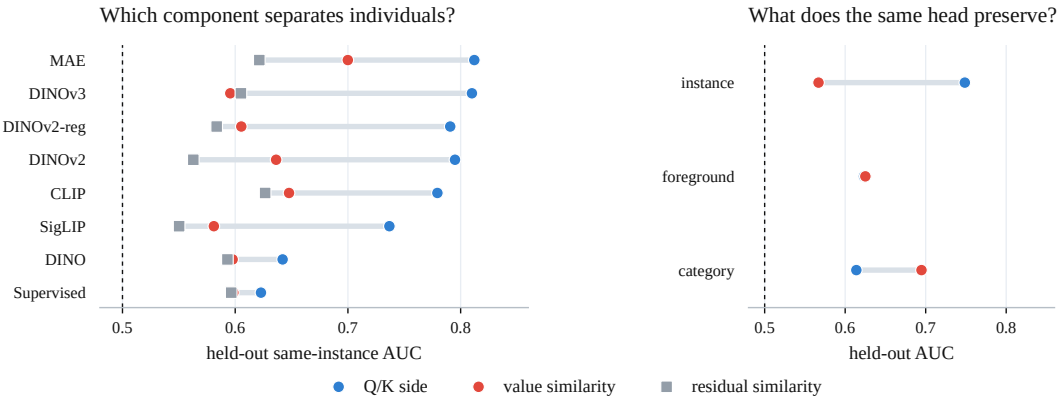

Figure 3: **Instance-sensitive heads separate routing from content.** Left: Q/K, value and residual comparisons each use their own development-selected component. Q/K separates individuals best in every backbone. Right: the same heads on two further pair tests, foreground versus background patches and one category versus another (Appendix A). Q/K better preserves instance membership while values better preserve category. Foreground-pair scores are similar on average.

a layer and head are chosen on the development split and then applied unchanged to held-out VOC and to COCO. Appendix A gives the exact protocol.

3.3 Routing distinguishes individuals while values preserve category

Results show that the routing side separates individuals better than value or residual similarity in every backbone (Fig. 3, left). The complete breakdown confirms that query similarity, key similarity, the query–key relation and attention all expose individuation more strongly than residual or value similarity (Table 3). Head rankings transfer from VOC to COCO (Spearman .93–.98).

Figure 3(right) reveals a more pronounced division. At the same instance-sensitive heads, the routing side is the strongest for *which individual* a patch belongs to, while V is the strongest for *what category* that individual belongs to. This goes beyond a generic "where versus what" distinction (Zhang et al., 2024): relative position is matched, yet the routing side still keeps zebra A separate from zebra B while the values preserve that both are zebras.

Q, K, and V are projections of the same residual state, so this separation is not automatic. Residual similarity reaches only .55–.63 AUC across models, compared with .62–.81 for the best held-out routing-side relation. The instance signal is therefore much easier to see inside selected attention heads than in ordinary residual similarity. The instance signal concentrates in particular heads and layers. In most backbones, a small set of heads stands out and the rest are much weaker (Ap-

pendix B). In DINOv2, DINOv3 and MAE the strongest heads lie in the last layers. In the supervised model, no head stands out. This localization is why we study one DINOv2 head in detail below.

4 FOLLOWING WHERE INSTANCE-SENSITIVE HEADS READ

What does an instance-sensitive head actually do? We now follow where each patch reads, with the same experimental setup as in Section 3 and the heads selected there kept frozen. We compare full attention patterns, summarize each pattern by one destination, and test what this destination tracks.

4.1 SHARED READING AND THE ATTENTION ADDRESS

Direct attention considers whether two patches attend to one another. The *full attention pattern* of a patch is its row of A, its weights over all destinations. Do patches from the same individual *read the rest of the scene in the same way*? The head and leg of zebra A can look less alike than the heads of zebras A and B, yet both parts of zebra A can place attention on the same regions. Similarity of full attention patterns distinguishes same-instance pairs better than direct attention in nearly every model and dataset we tested (Appendix C, Table 5). *Reading from similar destinations is a stronger same-instance signal than attending to one another.*

For each patch, we summarize where its attention lands by its mean destination. We call this its *attention address*. Let p_j be the (row, column) position of patch j on the 14×14 grid. Attention can also land on prefix or register tokens (Darcet et al., 2024), which have no image position, so we renormalize it over the image patches $\mathcal{P}$: $\widetilde{A}_{ij} = A_{ij} / \sum_{k \in \mathcal{P}} A_{ik}$. The address of patch i is

$$\mu_i = \sum_{j \in \mathcal{P}} \widetilde{A}_{ij} p_j. \tag{2}$$

Drawing an arrow from p_i to μ_i for every patch turns the head into a visible *address field* (Figs. 1 and 4). Patches from one object often point toward one region, while patches from another point elsewhere. The address is not necessarily the object center or the location of a sharp attention peak. It is the mean destination of the full attention pattern, which may contain several peaks.

4.2 THE ADDRESS FOLLOWS THE INDIVIDUAL, NOT ITS APPEARANCE

If the address belongs to an individual rather than to its appearance, different-looking parts of one individual should share it, and identical copies should not. We tested these hypotheses in DINOv2.

Different-looking parts share an address. In COCO, images containing several people, two different body parts of the same person, such as a head and a torso, form a positive pair. The same two parts on different people form a negative, with displacement matched as before (Appendix C.3). Despite their different appearance, the addresses distinguish individuals at .935 AUC, close to the full attention pattern and well above residual, value, and query similarity (Table 1, right).

Identical copies receive different addresses. If the address merely described appearance, identical copies would collide. We place pixel-identical copies of a segmented zebra on a natural background. Each copy receives its own address μ. When one copy moves, its address follows it, while the address of the stationary copy barely changes (Fig. 4). Li et al. (2025) find that their probe separates identical objects in middle layers but not near the top of the network. Here, late-layer heads keep the copies apart, at layer 10 of 12 in both DINOv2 and DINOv3 (Appendix C.4).

The address thus belongs to the individual rather than to its appearance. Why does the head produce such shared destinations for each instance?

5 HOW A QUERY SPECIFIES WHERE TO READ

The attention address summarizes where attention lands once the head has compared a query with every key in the image. It shows where a patch reads, not how the query–key computation chose that destination. *How can patches that look different end up reading from the same region?* Each

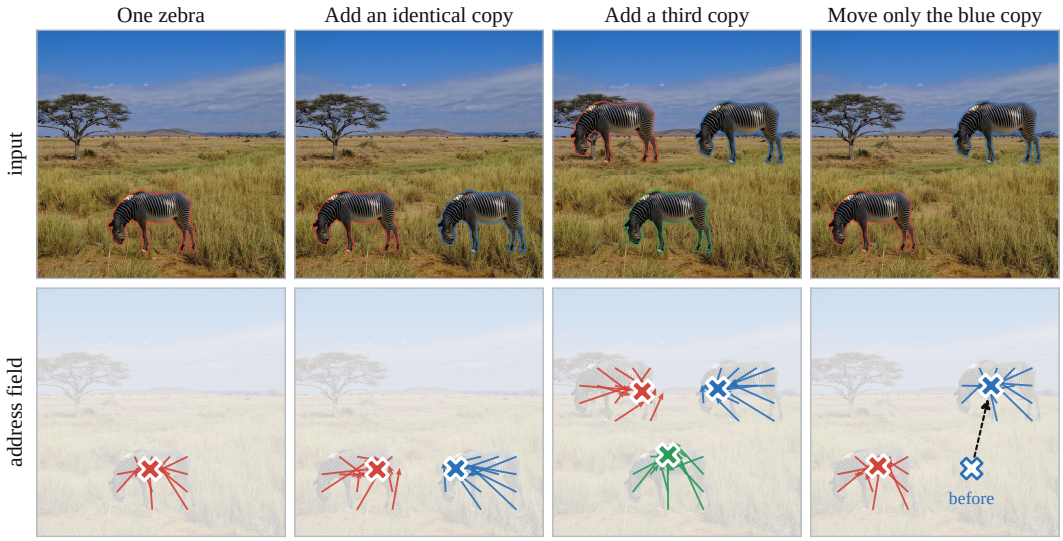

Figure 4: **Identical appearance, separate addresses.** Pixel-identical copies of one zebra are placed on the same background (top). The patches of each copy point to their own destination (bottom; crosses mark the mean address of each copy). When only the blue copy moves, its address follows it, while the address of the red copy barely changes.

query compares against a key at every image location. One simple hypothesis is that, for a patch on zebra A, these scores rise toward one region of the image and fall away from it, like a smooth peak, whatever that region contains. Scores of this kind would arise if part of each key encoded where its patch lies, and the query weighted this part to select a region. We first look for such a spatial pattern in the keys, then describe how each query uses it (Section 5.1). Because this description belongs to the query, we can compare it with the observed address (Section 5.2) and edit it (Section 6).

We focus our study on the DINOv2 (Oquab et al., 2024) head selected by the controlled test of Section 3.2, layer 10, head 2 (L10h2), with the same development and held-out splits. Each patch on its 14×14 grid has a 64-dimensional query and key.

5.1 From a spatial pattern in the keys to a spatial request

Keys contain a simple spatial pattern. A key depends on both what is in the image and where its patch lies. To see what stays tied to location, we averaged keys from the same patch position across development images. Differences between images are reduced, while patterns that recur at the same positions remain. Content that often appears in one part of the image also affects this average. We therefore examine only whether a simple function of position describes the average key well. The form of this function matters because a query scores each location through the dot product with the key there: for a patch on zebra A to favor zebra A, these products must be higher on zebra A than elsewhere. A key that varies linearly with position lets a query favor only one side of the image. A quadratic form can also favor a region between the edges, such as one zebra, and is the lowest order that can do so. After the softmax, a quadratic score can form a Gaussian-shaped peak of attention anywhere in the image. With r and c the row and column coordinates of the patch grid, centered and scaled, the average key at (r, c) is

$$\bar{k}(r, c) \approx b_0 + r\, b_r + c\, b_c + r^2 b_{r^2} + c^2 b_{c^2} + rc\, b_{rc}. \tag{3}$$

Each coefficient b is a 64-dimensional vector, fitted by least squares from patch coordinates and average keys only, then kept fixed. No object labels are used, the ViT is unchanged, and no new module is added. We call these vectors the *spatial basis*. The quadratic function captures most, but not all, of the variation of the mean key across positions: 91% on the development split, against 67% for a linear fit. A cubic fits slightly better without a consistent gain in routing or grouping (Appendix D, Table 7).

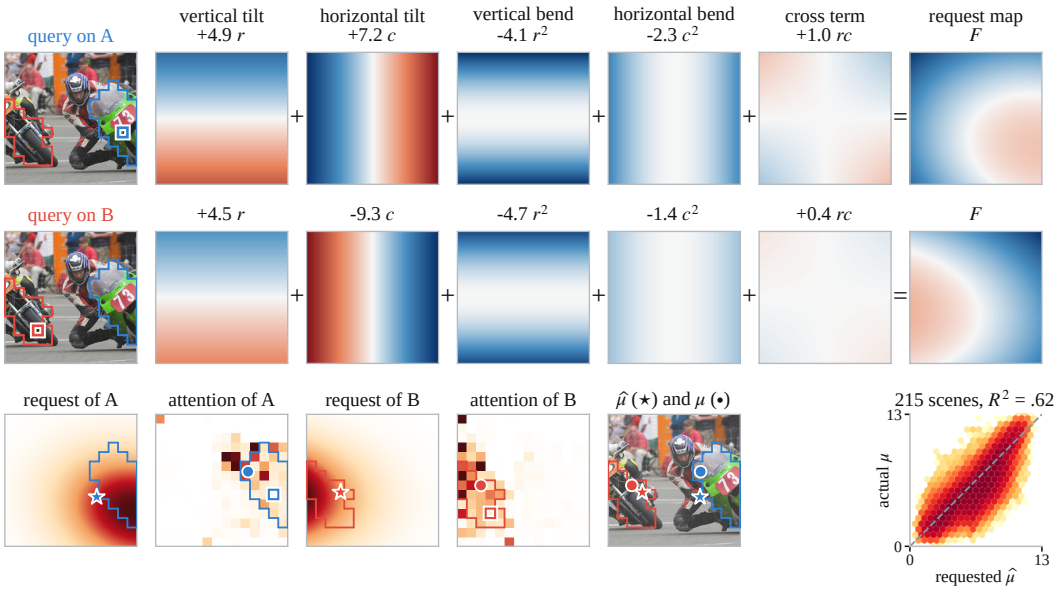

Figure 5: **A query builds its spatial request from five position maps.** Top rows: for one query on each motorcyclist, the request θ weights the five position maps of the spatial basis, and their sum is the request map F (Eq. 4). Bottom: the softmax of F gives the request distribution $\widehat{A}$, whose mean is the requested destination $\widehat{\mu}$ ($\star$). The actual attention of the same query reads nearby, with address μ ($\bullet$). Bottom right: across all patches of 215 COCO scenes, requested and actual destinations track one another. The dashed line is exact agreement.

The query weighs this pattern. If we replace each key by its spatial description, the score of query i at location (r, c) becomes $q_i^\top b_0 + r\, q_i^\top b_r + c\, q_i^\top b_c + \dots$. The first term is the same for every destination and cancels in the softmax. The query thus interacts with the spatial pattern through five coefficients. Let B be the 5×64 matrix whose rows are the five nonconstant basis vectors. We call these five coefficients, $\theta_i = Bq_i$, the *spatial request* of patch i, or its request for short. They define the *request map* F_i, the score that the query gives each location through the spatial basis alone:

$$\begin{aligned} \theta_i &= (\alpha_i, \beta_i, \gamma_i, \delta_i, \epsilon_i) = Bq_i, \\ F_i(r, c) &= \alpha_i r + \beta_i c + \gamma_i r^2 + \delta_i c^2 + \epsilon_i rc. \end{aligned} \tag{4}$$

The two linear terms tilt this map across rows and columns, and the other three bend it (Fig. 5). The softmax of the head turns F_i into a distribution over image patches, $\widehat{A}_i$. Intuitively, $\widehat{A}_i$ shows where patch i would read if the keys carried only their spatial pattern: a smooth region, which in Fig. 5 lies on the rider of each query. We summarize it like the address, by its mean location, the *requested destination* $\widehat{\mu}_i$:

$$\widehat{A}_{ij} = \mathrm{softmax}_{j \in \mathcal{P}}\left(F_i(r_j, c_j)/\sqrt{d}\right), \qquad \widehat{\mu}_i = \sum_{j \in \mathcal{P}} \widehat{A}_{ij} p_j. \tag{5}$$

Address and request. The address μ_i is measured: where patch i reads in this image. The request θ_i is computed from the query and the fixed spatial basis: where the query would read if keys carried only their spatial pattern. Up to a constant that cancels in the softmax, every actual attention score is the request score F_i plus a term from key content that the basis omits, such as image-specific details (Appendix D). The request is therefore the spatial part of the native score, not a substitute for attention, and this remainder is why $\widehat{\mu}_i$ predicts μ_i only in part.

5.2 The request retains the instance relation and predicts the address

If the request is part of how L10h2 routes, it should keep the same-instance relation that selected this head, and its destination should predict where attention lands.

Across backbones, held-out COCO						*Different parts of one person*		
Backbone	Head	Original Q/K	Request θ	Requested $\widehat{\mu}$	Address μ	Readout	Dim.	AUC
DINO	L7h3	.623	.644	**.661**	.660	*Vector similarity*		
DINOv2	L10h2	.738	**.823**	.821	.807	Residual	768	.651
DINOv2-reg	L11h3	.730	.805	**.806**	.804	Value	64	.720
DINOv3	L10h1	.765	**.815**	.814	.801	Query	64	.856
MAE	L12h8	.765	.830	**.831**	.793	*Where attention goes*		
CLIP	L9h3	.746	**.769**	.747	.707	Attention address μ	2	.935
SigLIP	L9h2	.702	**.720**	.710	.627	Requested destination $\widehat{\mu}$	2	.940
Supervised	L11h10	.609	**.614**	.602	.581	Full attention pattern	196	**.945**

Table 1: **Where attention goes separates individuals.** *Left:* same-instance AUC at the head selected by the original test in each backbone (Section 5.2). Original Q/K is the routing-side relation selected there, and bold marks the best entry in each row. *Right:* different parts of one person (Section 4), the two-coordinate address and requested destination nearly match the full attention pattern. Dim. is the dimension of the compared vectors.

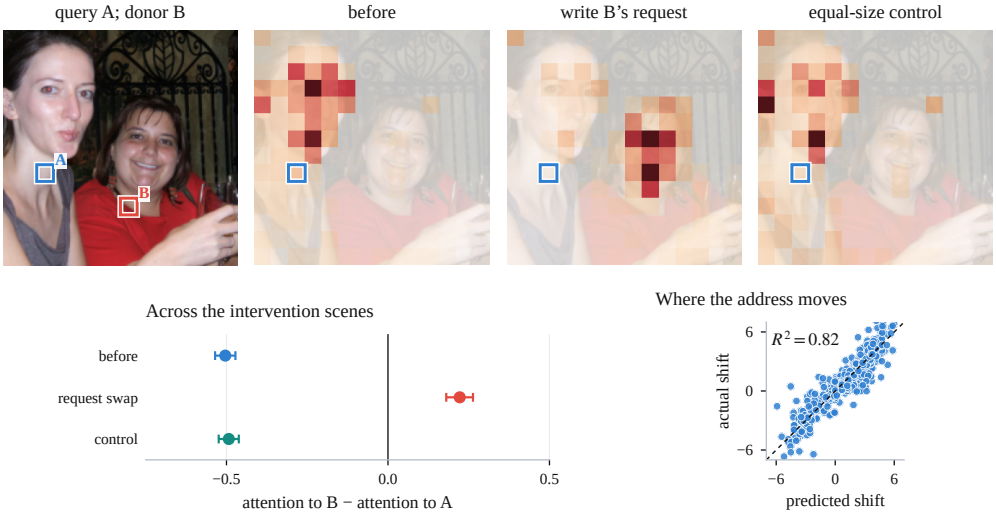

Figure 6: **Changing the spatial request redirects attention.** Top: a torso patch on person A reads from A. After the request swap, it reads from B, and after the equal-size control, it stays on A. Maps share one color scale, with square-root contrast. Bottom, over all 215 scenes: attention to B minus attention to A (left, from -1 for only A to $+1$ for only B) and actual versus predicted address shift (right).

On the matched patch pairs of Section 3.2, the requested destination $\widehat{\mu}$ separates individuals. On held-out VOC it reaches .872 AUC, above .795 for direct attention, and the advantage persists on COCO. It falls close to chance in an untrained network or when requests come from another image (Appendix D). At the selected head of every backbone, the request matches or improves on the original Q/K relation, and the requested destination separates individuals at least as well as the observed address (Table 1, left). The separation that the address showed in Section 4 is thus already present in this component of the query.

In Figure 5, the requested destination of each query lies near its observed address. Across all patches of 215 COCO scenes, $\widehat{\mu}$ explains 62% of the variation in μ (Fig. 5, bottom right). The comparison is direct: $\widehat{\mu}$ is computed without the attention map, and nothing is fitted between the two. The agreement is partial, particularly on object patches, because attention also responds to image-specific key content that the request omits. At the selected head of five of the other seven ViT-B backbones, the requested destination predicts the address as well ($R^2 = .57$–$.84$, Appendix E.2).

6 EDITING AND GROUPING WITH THE SPATIAL REQUEST

The address can only be observed. The request, in contrast, is read from the native query of the head along five fixed directions, $\theta_i = Bq_i$, so moving the query along them changes the request while the model stays unchanged. We use this to edit the request and to group patches with it.

6.1 CHANGING THE REQUEST REDIRECTS ATTENTION

A request that predicts the address could still be only a correlate of routing. To test whether it controls where the head reads, we took a source patch on person A and a donor patch on person B in each of the 215 COCO scenes, and compared two edits of the source query at L10h2 (Fig. 6). The *request swap* gives the source query the request of the donor, using the smallest query change that does so. The *equal-size control* changes the query by the same amount in random directions that leave its request unchanged. In both cases we edited only this query at this head, kept the image fixed, and reran the model.

After the request swap, attention shifts from A toward B, while after the equal-size control the original route stays largely unchanged. The predicted shift of the requested destination explains 82% of the variation in the actual address shift, again without fitting anything between the two. Changing this component of the native query therefore changes where the head reads, and attention moves gradually as the edit is scaled. The same swap reverses the preference of the selected head in six of the other seven ViT-B backbones and in the larger DINOv2 ViT-L, and in all of these except SigLIP the requested destination also predicts the movement (Appendix E.2). Later instance-sensitive heads respond to this large edit but still prefer person A (Appendix E), so the edit controls where this head reads, not an identity carried through the network.

6.2 GROUPING REPEATED OBJECTS

If requests carry instance structure, a simple frozen grouping of them should separate repeated objects in a whole image. For each image, we clustered its patches by their request vectors θ_i with mean-shift, which does not need the number of objects. Its one parameter, a radius, is selected for each representation on labeled VOC development images and then frozen, and no segmentation model is trained (Appendix F). We score instance assignment only among ground-truth foreground patches, with the foreground adjusted Rand index (FG-ARI).

On all 1,587 eligible COCO scenes, request grouping reaches $.615$ FG-ARI, versus $.289$ for the same clustering of final DINOv2 patch features and $.356$ for the object-centric model SPOT (Kakogeorgiou et al., 2024). Full attention patterns reach $.649$. Requests therefore better separate repeated instances that feature-based grouping tends to merge, in seven of eight backbones (Appendix F).

7 CONCLUSION

We studied how a pretrained ViT keeps two objects of the same category apart. With category and position controlled, selected heads separate them through where patches read more than through what they carry. Different-looking parts of one person read from a common place, their attention address, while identical copies read from different places. Part of each query, its spatial request, asks for a region of the image. In DINOv2 and most other backbones, this request predicts the address, and changing it redirects the head to the other object. Patches can thus agree on an object by reading from the same place, without looking alike.

Limitations and future work. Requests separate repeated foreground instances well, but they do not by themselves recover complete object masks: background regions remain fragmented and object boundaries are imperfect. A natural direction is to identify complementary mechanisms for foreground/background separation and object boundaries, and combine them with instance-sensitive routing for stronger object discovery. Another is to test whether pretrained video models use similar routing to maintain object identity over time.

AI USE STATEMENT

In this work, we used generative AI tools to help implement and run experiments and analyze their results, to search for related work, and to draft and polish text. The authors independently checked the reported results and are responsible for the final claims and submission.

REFERENCES

Rim Assouel, Declan Campbell, Yoshua Bengio, and Taylor Webb. Visual symbolic mechanisms: Emergent symbol processing in vision language models. In *International Conference on Learning Representations*, 2026.

Anil Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406, 1946.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. On attention redundancy: A comprehensive study. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 930–945, 2021. doi: 10.18653/v1/2021.naacl-main.72. URL https://aclanthology.org/2021.naacl-main.72/.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning*, pp. 2286–2296. PMLR, 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.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *International Conference on Learning Representations*, volume 2024, pp. 36391–36413, 2024.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Lianghuan Huang, Yihao Li, Saeed Salehi, Yingshan Chang, Ansh Soni, and Konrad P Kording. Formalizing the binding problem. In *International Conference on Machine Learning*, 2026.

Daniel Kahneman, Anne Treisman, and Brian J Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive psychology*, 24(2):175–219, 1992.

Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22776–22786, 2024. doi: 10.1109/CVPR52733.2024.02149.

Raphi Kang, Hongqiao Chen, Georgia Gkioxari, and Pietro Perona. Linear mechanisms for spatiotemporal reasoning in vision language models. In *International Conference on Learning Representations*, volume 2026, pp. 41245–41258, 2026.

Bum Jun Kim, Hyeyeon Choi, Hyeonah Jang, and Sang Woo Kim. Understanding gaussian attention bias of vision transformers using effective receptive fields. In *British Machine Vision Conference*, 2023.

Yihao Li, Saeed Salehi, Lyle Ungar, and Konrad Kording. Does object binding naturally emerge in large pretrained vision transformers? *Advances in Neural Information Processing Systems*, 38: 3394–3423, 2025.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in neural information processing systems*, volume 33, pp. 11525–11538, 2020.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5401–5411. IEEE, 2025.

Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8354–8365. IEEE, 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.

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

Zenon Pylyshyn. The role of location indexes in spatial perception: A sketch of the finst spatial-index model. *Cognition*, 32(1):65–97, 1989.

Samyak Rawlekar, Amitabh Swain, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Finding distributed object-centric properties in self-supervised transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31326–31335, 2026.

Khalil Sabri, Guillaume-Alexandre Bilodeau, Nicolas Saunier, and Wassim Bouachir. Semantic slots for video object-centric learning. In *British Machine Vision Conference*, 2026. arXiv:2608.21636.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2023.

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 *British Machine Vision Conference*, 2021.

Alexa R Tartaglini and Michael A Lepori. I walk the line: Examining the role of gestalt continuity in object binding for vision transformers. *arXiv preprint arXiv:2604.09942*, 2026.

Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14523–14533. IEEE, 2022.

Xiao Zhang, David Yunis, and Michael Maire. Deciphering 'what'and 'where'visual pathways from spectral clustering of layer-distributed neural representations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4165–4175. IEEE, 2024.

A DATA, SELECTION, AND THE CONTROLLED TEST

VOC development images are used to select the components, fit the spatial bases and choose the grouping radii. Disjoint VOC images are used to evaluate those choices. Each COCO test uses every validation image that meets its annotation requirements (Table 2). Parts need visible human keypoints, and grouping needs reliable instance masks.

Image set	Size and role	Experiments using it
VOC development	102 images; 62,362 pairs	Select layers, heads and relations; fit spatial bases without object labels; select grouping radii
VOC held-out	101 images; 71,390 pairs	Native relations, shared reading, spatial requests and controls
COCO pair test	1,414 images; 938,606 pairs	Transfer the same frozen native relations and spatial bases
ADE20K pair test	530 images; 176,732 pairs	Transfer the native routing–value comparison
COCO parts	215 images; 35,100 pairs	Compare different body parts; predict addresses; intervene once per image with five controls
COCO grouping	1,587 images	Group all patches; evaluate instance assignments and object masks
Synthetic copies	Three zebra cutouts on three backgrounds	Separate appearance from identity and position
DAVIS illustrations	Fixed qualitative examples	Show grouping, retrieval and resolution effects

Table 2: **One controlled question, with specialized tests where needed.** Pair counts include both labels and count each sampled pair once. The COCO tests are overlapping views of one validation population, not independent replications. DAVIS frames illustrate grouping and resolution effects.

A.1 WHICH IMAGES AND PATCHES ENTER THE TEST?

We used Pascal VOC 2012 validation images for development and internal evaluation. A fixed 256-image repeated-category list contains 203 images that support the controlled pair test of Section 3.2. Alternating entries define the 102-image development and 101-image held-out splits, before any component is selected. All spatial fits use the same 102 development images, and all grouping settings are chosen within that split.

An image must contain two instances of one category, each with at least two labeled patches, and at least eight foreground patches overall. A patch receives an instance label when that instance occupies at least 60% of its nonignored pixels. Ambiguous overlapping pixels are ignored, and COCO crowd regions do not count as individual objects. Same-instance pairs are positives, and pairs from different instances of one category are negatives. We matched the horizontal and vertical displacement, including direction: for every included offset, we sampled the same number of positive and negative pairs. We kept at most 12 pairs per label at each offset, so common offsets do not dominate, and required at least 24 pairs per image. These fixed sampling rules determined the counts in Table 2. Every COCO image that meets them is used, and no image is removed because of how a model scores on it. The parts, copy and grouping tests use their own rules (Appendices C.3, C.4 and F).

A.2 HOW ARE MATCHED PAIRS SCORED?

Matching makes each relative offset equally likely under either label, so distance or direction alone cannot solve the test. We evaluated the scores below on these matched pairs.

Each readout gives every pair one score, and a higher score should indicate the same instance. Query, key, value and residual similarities are cosines. Residuals are the patch representations after a transformer block, and their layer is selected separately, by the highest mean same-instance AUC

on VOC development. For the query–key relation and direct attention, we averaged both directions because same-instance membership is symmetric:

$$s_{ij}^{Q\leftrightarrow K} = \tfrac{1}{2}(q_i^\top k_j + q_j^\top k_i), \qquad s_{ij}^{A\leftrightarrow} = \tfrac{1}{2}(A_{ij} + A_{ji}). \tag{6}$$

Addresses, requested destinations and spatial requests use negative Euclidean distance, and full attention patterns use the overlap of the two attention rows (Appendix C). Each image supplies one AUC, the probability that a random positive pair scores higher than a random negative pair, so no decision threshold is needed. Every image receives equal weight. Confidence intervals use 5,000 resamples of images, not of individual patch pairs.

A.3 Which native relations carry the advantage?

Table 3 expands the comparison of Figure 3. For each relation, a layer and head were selected on VOC development images and then used unchanged on both held-out sets. No choice uses test outcomes.

Set	Backbone	Residual	Query	Key	Query–key	Attention	Value
VOC	DINO	.593	**.644**	.620	.642	.627	.598
	DINOv2	.563	.740	.734	.741	**.795**	.636
	DINOv2-reg	.584	.749	.770	.786	**.791**	.605
	DINOv3	.605	.801	.747	.782	**.810**	.596
	MAE	.621	.777	.764	.794	**.812**	.700
	CLIP	.627	.715	.722	.755	**.779**	.648
	SigLIP	.550	.655	.650	.727	**.737**	.581
	Supervised	.596	.623	.605	.613	**.648**	.598
COCO	DINO	.586	**.623**	.606	.623	.608	.588
	DINOv2	.565	.707	.693	.690	**.738**	.616
	DINOv2-reg	.565	.700	.722	**.732**	.730	.589
	DINOv3	.602	**.765**	.725	.746	.765	.606
	MAE	.610	.725	.699	.742	**.765**	.659
	CLIP	.611	.703	.700	.728	**.746**	.631
	SigLIP	.562	.625	.620	.693	**.702**	.584
	Supervised	.594	.609	.612	.609	**.625**	.592

Table 3: **The routing advantage appears in several native relations.** Mean same-instance AUC on 101 VOC and 1,414 COCO images. Bold marks the strongest routing relation within each row. All choices use the development set.

The right panel of Figure 3 asks a complementary question: at the same instance-sensitive heads, what do the routing side and the values preserve best? It compares three pair tests on the 101 held-out VOC images, each with displacement matched. The instance test is the one of Section 3.2. In the foreground test, a positive pair contains two foreground patches and a negative pair contains one foreground and one background patch. In the category test, both patches of every pair come from different instances, and only positive pairs share a category. All 101 images support the foreground test and 47 support the category test. Background here means no reliable foreground assignment, which includes mixed boundary patches, so this row compares patch pairs and is not a foreground segmentation.

A.4 Does the result extend beyond the small VOC split?

The development split is small on purpose, 102 VOC images. If instance-sensitive routing is already part of how a pretrained ViT works, a few images should be enough to find the heads that carry it, and those heads should then work unchanged on larger datasets. They do. With every choice frozen on VOC, routing exceeds value similarity in all eight backbones on the 1,414-image COCO test (Figure 7). On the 530-image ADE20K test, routing also exceeds value similarity in every backbone.

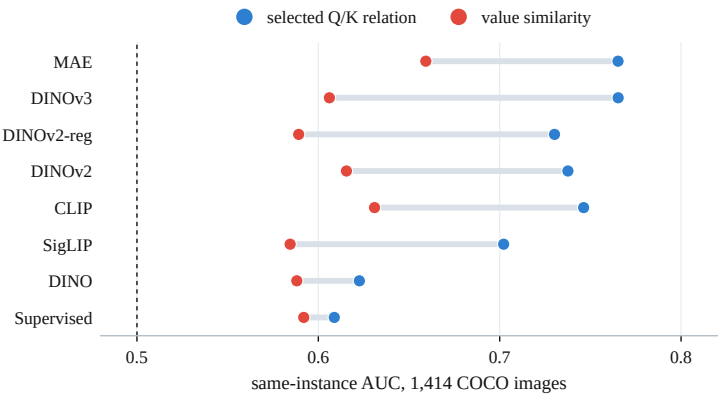

Figure 7: **Routing remains stronger than value similarity on the larger COCO test.** Blue and red show the actual AUCs of the independently VOC-selected routing and value components.

A.5 Model inputs and reproducibility

All backbones are frozen public *timm* (Python Image Models) checkpoints in evaluation mode (Table 4). Layers are numbered from 1 and heads from 0. Inputs are resized on the shorter side and center-cropped to a square: 196 pixels for patch-14 models and 224 for patch-16 models, giving 14×14 patches in both cases. Image resizing is bilinear and masks use nearest-neighbor resizing. Inputs use the channel normalization of each checkpoint. Learned absolute position embeddings are interpolated by *timm* when the grid changes, and DINOv3 uses its native rotary positions. DAVIS illustrations resize the full frame to a square. The higher-resolution examples use 392-pixel DINOv2 inputs and a 28×28 grid with the same basis and radii.

Q, K and V are captured at scaled dot-product attention, after the Q/K normalization of the architecture and the DINOv3 rotary embeddings. Prefix and register tokens take part in native attention. Spatial distributions then discard those destinations and renormalize over image patches, both for the attention address and for the requested destination.

Model	Checkpoint
DINO	vit_base_patch16_224.dino
DINOv2	vit_base_patch14_dinov2.lvd142m
DINOv2-reg	vit_base_patch14_reg4_dinov2.lvd142m
DINOv3	vit_base_patch16_dinov3.lvd1689m
MAE	vit_base_patch16_224.mae
OpenCLIP	vit_base_patch16_clip_224.openai
SigLIP	vit_base_patch16_siglip_224.webli
AugReg	vit_base_patch16_224.augreg_in21k_ft_in1k

Table 4: Frozen backbones used in the eight-model comparison.

B WHERE DOES INSTANCE-SENSITIVE ROUTING APPEAR?

The main comparison selects components on development images. Figure 8 shows the surrounding heads on the held-out set: strong instance discrimination is concentrated in a few heads rather than spread across attention. Figure 9 shows how the address field changes with depth in one fixed scene. The most instance-discriminative head can change with the scoring convention, so localization does not imply a unique head.

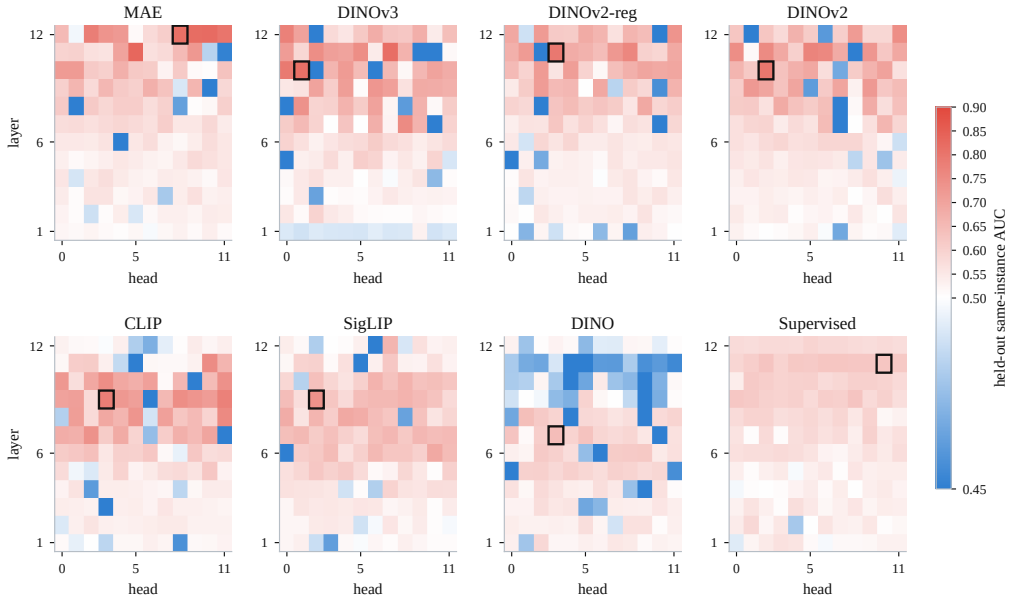

Figure 8: **Instance-sensitive routing is localized.** Each cell is the held-out same-instance AUC of one layer and head. The outlined square marks the component selected on the development split. Strong heads appear at different depths across pretraining methods, and most heads have weaker instance discrimination.

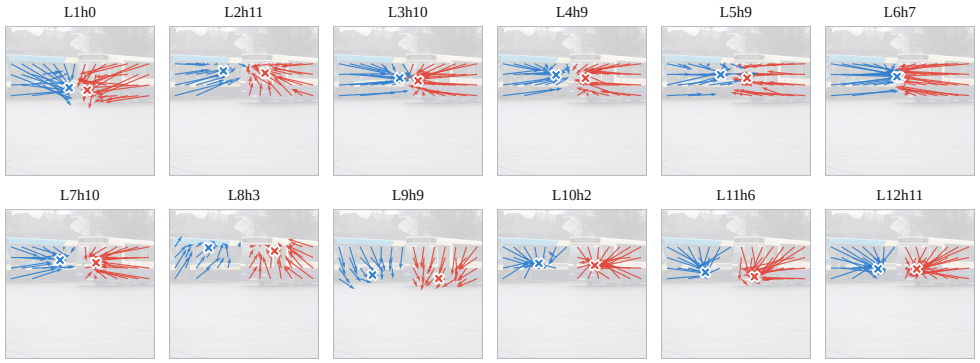

Figure 9: **Object-specific address fields emerge through depth.** At each layer the head is selected by direct-attention AUC on the same development split. All panels show the same held-out scene. Later DINOv2 heads form two compact destinations, but the pattern is neither universal nor monotonic.

C What does the attention address capture?

This appendix supports Section 4. It compares full attention patterns with direct attention, tests whether locality alone explains the addresses, and describes the two tests of what the address tracks: different parts of one person and identical copies.

C.1 Is shared reading more informative than direct attention?

We compared the full attention patterns of two patches with the Bhattacharyya coefficient, $\sum_k \sqrt{\widetilde{A}_{ik}\widetilde{A}_{jk}}$ (Bhattacharyya, 1946). This classical overlap measure has also been used to compare attention vectors across BERT heads (Bian et al., 2021). Here it compares different patches within one head. Table 5 keeps the head selected by the original instance test fixed. Full attention patterns (shared reading) separate instances better than direct attention in 14 of 16 model–dataset comparisons.

	Held-out VOC			COCO		
Backbone	Full attention pattern	Direct attention	Gain	Full attention pattern	Direct attention	Gain
DINO	.667	.627	+.041	.631	.608	+.023
DINOv2	.873	.795	+.078	.818	.738	+.081
DINOv2-reg	.853	.791	+.063	.792	.730	+.062
DINOv3	.852	.810	+.042	.812	.765	+.047
MAE	.833	.812	+.021	.770	.765	+.005
CLIP	.789	.779	+.010	.756	.746	+.009
SigLIP	.705	.737	−.032	.675	.702	−.027
Supervised	.636	.495	+.141	.610	.514	+.096

Table 5: **Full attention patterns strengthen the same-instance relation.** Same-instance AUC at the head selected by the original test in each backbone, on the same matched pairs.

C.2 Could local attention explain the addresses?

Could nearby patches share an address simply because attention stays nearby? We built a distance-only attention map for each query: nearby destinations receive more weight than distant ones, regardless of the image. We set its Gaussian width so that attention travels the same average distance as in the real head. A second control uses the attention maps that the head produces for another image, placed on the current grid. This keeps realistic reading patterns while breaking their connection to the current objects.

Both controls use the original 101 held-out images, selected heads and matched patch pairs. For reproducibility, each image borrows attention from the previous image in the saved list, with the first borrowing from the last. We compared patches by the distance between their resulting addresses. In DINOv2, real addresses reach .870 AUC, versus .518 for distance-only attention and .523 for attention from another image. The real addresses outperform both controls in all eight backbones (Table 6). They therefore reflect the current scene, beyond nearby patches tending to read nearby. Figures 17 and 18 show visual examples.

Address from	DINO	DINOv2	DINOv2-reg	DINOv3	MAE	CLIP	SigLIP	Supervised
Real attention	**.708**	**.870**	**.859**	**.847**	**.876**	**.744**	**.636**	**.616**
Distance only	.523	.518	.520	.518	.514	.519	.507	.509
Another image	.520	.523	.516	.529	.505	.524	.508	.499

Table 6: **Locality does not explain the addresses.** Same-instance AUC of the attention address on the 101 held-out VOC images, at the head selected in each backbone.

C.3 Different parts of the same person

Can two visually different parts of the same person share a destination? COCO keypoints let us test this while keeping the original same-instance comparison. We assigned patches to head, torso, arm or leg using visible keypoints: head and face keypoints define the head, shoulders and hips the torso, elbows and wrists the arms, and knees and ankles the legs. To avoid ambiguous assignments, patches must have at least 90% instance purity, contain no annotated overlap, and lie within 2.5 patches of the nearest visible keypoint. These rules depend only on annotations, not on model scores. The labels approximate parts from keypoints and are not dense part masks.

Both positive and negative pairs connect different body parts. Within each image, we balanced the labels at every ordered part pair and exact signed offset, keeping up to 24 of each label per group, and required at least 12 pairs. Scanning all COCO validation images yields 215 scenes and 35,100 pairs. The same scenes support the intervention of Appendix E. On these pairs the request nearly matches the full attention pattern (.942 versus .945 AUC).

C.4 Identical copies

The copy test of Figure 4 removes appearance as an explanation for different addresses. Three complete COCO zebra cutouts were each placed on three natural backgrounds, giving nine combinations. Within a scene the two copies are pixel-identical, and only patches covered by the object mask are scored. All nine combinations yield separate, correctly ordered, movable addresses in DINOv2 (L10h2) and DINOv3 (L10h1). When one copy moves, its address shifts by 4.68 and 4.61 patches on average, respectively, while the address of the stationary copy shifts by $.98$ and 1.22. Figure 4 uses a fourth background, licensed for reproduction, and the zebra with the median score of the three.

D How is the spatial request constructed and validated?

D.1 Fitting the spatial basis

We fitted the average key at each patch position using six maps, $\phi = (1, r, c, r^2, c^2, rc)$. On the 14×14 grid, r and c are the row and column coordinates, each centered and divided by its standard deviation. Keys are averaged at each position over the 102 development images. We find one key vector b_t for each map ϕ_t so that their weighted sum reconstructs these average keys:

$$\{b_t\}_{t=0}^{5} = \arg\min_{\{u_t\}} \sum_j \left\| \bar{k}(r_j, c_j) - \sum_{t=0}^{5} \phi_t(r_j, c_j) u_t \right\|_2^2, \tag{7}$$

where j indexes patches and each u_t has 64 entries. The constant vector b_0 cancels in the softmax. The other five vectors form the rows of the 5×64 matrix B of Section 5.1, so the five entries of $\theta_i = Bq_i$ are the dot products of the query with these vectors. We froze the fitted vectors and reused them for evaluation and illustrations. Labels selected the head, but the fit itself uses only coordinates and native keys. Figure 5 shows how a query weights the five maps to produce its request.

D.2 Why use a quadratic function if two coefficients already separate instances?

Two slopes already separate instances well. Why keep curvature? Slopes can only tilt attention toward one side of the image, while curvature lets a query place a peak of attention around one region, such as its own object (Section 5.1). It also helps explain where attention goes, how it moves under edits, and which patches group together. Table 7 compares linear, quadratic and cubic descriptions of the same head.

	Key fit	Same-instance AUC ↑		Address	Movement	Grouping ↑	
Spatial description	R^2 ↑	VOC	Person parts	error ↓	R^2 ↑	FG-ARI	mIoU
Linear	.669	.871	.909	2.11	.646	.556	.301
Quadratic	.907	**.872**	**.940**	**1.90**	.821	**.615**	**.429**
Cubic	**.941**	.869	.939	1.92	**.827**	.594	.406

Table 7: **Curvature helps explain routing and grouping.** Key fit is the share of the variation in the development mean keys that each description explains. Same-instance AUC compares requests on held-out VOC and requested destinations on different parts of one person (Appendix C.3). Address error, in patches, and movement R^2 use the 215 part scenes and the request swaps of Appendix E, with nothing fitted to the outcomes. Grouping uses the rule of Appendix F on all 1,587 COCO images. Bold marks the best value in each column.

Curvature barely changes pairwise discrimination on VOC, but it improves the address prediction, the movement prediction and grouping. Cubic terms fit the average keys better still, but bring little change in routing prediction and no gain in grouping. These results favor the quadratic as a compact description.

D.3 Does the request retain the instance relation?

Table 8 supports Section 5.2. At the DINOv2 L10h2 head, the spatial request and the requested destination retain the same-instance relation on both held-out sets.

	VOC	COCO
Residual similarity	.563	.565
Query similarity	.740	.707
Direct attention	.795	.738
Attention address μ	.870	.807
Spatial request θ	.872	.823
Requested destination $\widehat{\mu}$	.872	.821
Full attention pattern	.873	.818
Slopes only	.871	.805
Curvature only	.628	.655

Table 8: **The spatial request retains the relation that selected the head.** Same-instance AUC on held-out VOC and COCO.

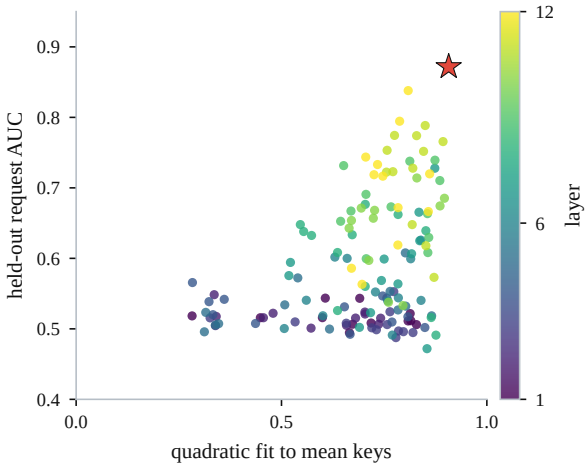

Figure 10: **Keys fitted with quadratic alone are not enough.** Each point is one of the 144 heads of DINOv2, with its own development fit and request AUC on held-out VOC. The selected head (star) has the strongest fit and request AUC, while many heads with quadratic keys carry little instance signal.

Three additional controls examine simpler explanations. *Architecture alone:* the same construction in a seeded untrained network gives request AUCs of .517 on VOC and .537 on COCO. *A fixed spatial pattern:* assigning requests from another image to the current grid gives .530 and .513. *Category imbalance:* also matching the actual category within each offset leaves .881 and .841, on the 95 VOC and 1,263 COCO images with enough balanced pairs. The signal therefore depends on learned features of the current scene and survives matching category as well as relative position.

Figure 10 details whether keys fitted with quadratic alone are sufficient. They are not: many heads have a good quadratic fit but little instance discrimination. Fit and request AUC correlate at .38 across all 144 heads but only .06 within layers, so depth explains much of their association.

D.4 How close is the requested destination to the attention address?

Table 9 compares the requested destination with the actual attention address, as the mean distance in patches over all 1,414 COCO images. Two controls are used to give reference errors. *Wrong image* borrows the requested destination at the same grid position from another image. *Source position* predicts that each patch reads at its own location. For every backbone, the current request predicts better than both controls, although the remaining error varies.

Backbone	Current request	Wrong image	Source position
DINO	1.32	1.95	2.88
DINOv2	1.74	2.59	3.36
DINOv2-reg	1.55	2.58	3.38
DINOv3	1.42	2.77	3.34
MAE	1.17	2.04	3.79
CLIP	1.79	2.65	3.10
SigLIP	2.76	3.27	4.57
Supervised	2.38	2.54	5.11

Table 9: **The current request predicts attention destinations better than spatial controls.** Mean Euclidean error, in patches, between each predicted destination and the attention address over all patches of 1,414 COCO images.

D.5 Why does the request not reproduce all of attention?

The spatial basis describes the average key at each position well, but the keys of one image also carry what that image contains. Let e_j be the part of the actual key left out by the fitted basis, and let $\phi_j = (r_j, c_j, r_j^2, c_j^2, r_j c_j)^\top$ contain the five position-map values. Then

$$k_j = b_0 + B^\top \phi_j + e_j, \tag{8}$$

$$q_i^\top k_j = q_i^\top b_0 + \underbrace{\theta_i^\top \phi_j}_{F_i(r_j, c_j)} + q_i^\top e_j. \tag{9}$$

The first term is constant over destinations and cancels in the softmax. The middle term produces the request distribution $\widehat{A}_i$. The last term reweights it according to the key structure that the fit omits:

$$\widetilde{A}_{ij} = \frac{\widehat{A}_{ij} \exp(q_i^\top e_j / \sqrt{d})}{\sum_{t \in \mathcal{P}} \widehat{A}_{it} \exp(q_i^\top e_t / \sqrt{d})}. \tag{10}$$

Here $\mathcal{P}$ is the set of image patches and $d = 64$ for this head. The remainder includes image-specific key variation and errors in the mean-key fit. Changing a query can change both its request and its interaction with the remainder. The request swap of Section 6.1 therefore establishes control by that query component. It does not isolate the quadratic score as the only cause of the attention shift.

E How is the request edited, and does the mechanism hold beyond DINOv2?

E.1 Changing one query

We wanted to change the part of a query that sets its spatial request. To do so, we took the request of a donor patch on person B and wrote it into the query of a source patch on person A, using the smallest query change that achieved this. Every one of the 215 part scenes contributed its first sampled different-instance pair as source and donor, and no scene or pair is selected for successful redirection. If B^+ is the pseudoinverse of B, the request swap is

$$\Delta q = B^+(\theta_B - \theta_A). \tag{11}$$

This sets the request of the source to θ_B with the smallest change to its query, on average .60 of the query norm. The equal-size controls are five random query changes of the same length that satisfied $B\Delta q = 0$, so the request stays unchanged, and we averaged their results within each image. Figure 6 uses a torso pair for illustration, and its aggregate plots use all 215 predetermined pairs. Figure 11 shows one edit in more detail.

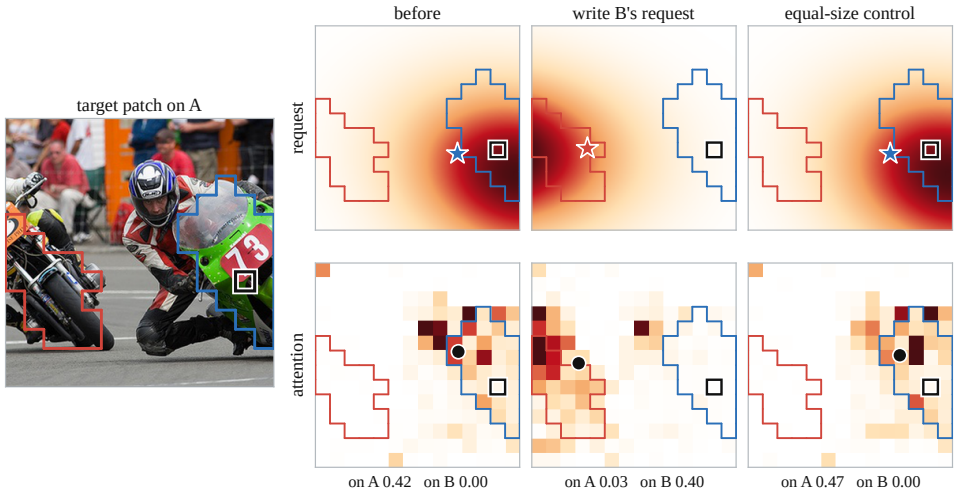

Figure 11: **The edited request moves where the patch reads.** Top: the request of one patch on rider A before the edit, after writing the mean request of rider B, and after an equal-size control edit that leaves the request unchanged. Bottom: the attention of that patch in the same three conditions, with its attention address (•) and the attention mass on each rider.

The requested destination predicts the direction of the address movement in 213 of 215 scenes. The address moves 3.68 patches on average, against 3.81 predicted and .40 under the equal-size controls. We measure the effect with B−A, the attention that the source patch places on the patches of person B minus the attention it places on person A, with attention renormalized over image patches. It ranges from -1 (the patch reads only from A) to $+1$ (only from B), so negative values mean more attention on A than on B, and positive values the reverse. The effect grows smoothly with the size of the edit: scaling Δq by .25, .5, .75, 1 and 1.5 moves B−A from $-.50$ to $-.40$, $-.21$, $+.03$, $+.22$ and $+.42$ (Figure 6 shows the full-size swap). Writing instead the request of another patch of person A, the one most different from the source, leaves attention on A ($-.41$). Random changes of the same size inside the space of requests move the address less predictably than the request of a real donor, so the prediction concerns the requests that patches actually make.

Condition	Edited L10h2	Selected L11h6	Selected L12h11
Intact	-0.504	-0.361	-0.280
θ swap	$+0.222$	-0.251	-0.242
Matched controls	-0.493	-0.360	-0.280

Table 10: **Editing the query component redirects this head.** Attention toward person B minus person A, averaged over all 215 part scenes. Later heads are selected for instance sensitivity on VOC development. Their attention shifts toward B but still favors A.

The later heads of Table 10 respond to the edit but keep their original preference. The result supports control of the reading of the edited head, not a lasting reassignment of identity.

E.2 Does the mechanism hold in other backbones?

We repeated the analysis at the development-selected head of every backbone, with its own frozen spatial basis, on the same 215 part scenes, pairs, donors and control seeds. We also ran DINOv2 ViT-L/14, whose head is selected by the same rule (L22h11). In most backbones, the DINOv2 pattern repeats (Table 11). The request separates different parts of one person better than query, value and final-feature similarity, and its destination predicts the address. The request swap reverses the preference of the head while equal-size controls do not, and the requested destination predicts how far the address moves. Two other DINOv3 heads that the rule selects in some re-splits, L11h7 and L12h0, give the same result.

Two backbones differ. In SigLIP, the swap still reverses the preference, but the movement is poorly predicted. Its head sends about a third of the attention of each patch to one single patch, an attention sink (Darcet et al., 2024), and SigLIP has no CLS or register token to absorb it. Once the two most-attended patches of each image are left out, the address and the movement are predicted again (R^2 of .60 and .65). In the supervised model, the selected head L11h10 shifts less and keeps its preference. This head is atypical: the same rule picks it in only 8% of random re-splits of the development images, and picks L9h8 in most of the others. At L9h8 the swap reverses the preference and the movement is predicted ($R^2 = .67$).

		Person parts AUC		Address	B − A		Movement	Grouping FG-ARI	
Backbone	Head	Request	Best other	R^2	Swap	Control	R^2	Request	Features
DINO	L7h3	.634	.625	.84	$-.24 \to +.09$	$-.23$	.84	.398	.194
DINOv2	L10h2	.942	.856	.62	$-.50 \to +.22$	$-.49$	.82	.615	.289
DINOv2-reg	L11h3	.949	.883	.72	$-.50 \to +.36$	$-.49$	.89	.585	.265
DINOv3	L10h1	.944	.860	.79	$-.48 \to +.15$	$-.47$	.88	.585	.380
MAE	L12h8	.942	.762	.76	$-.39 \to +.16$	$-.38$	.89	.617	.177
CLIP	L9h3	.895	.838	.57	$-.54 \to +.31$	$-.53$	.71	.511	.209
SigLIP	L9h2	.840	.734	$-.16$	$-.20 \to +.13$	$-.20$	$-.21$	.445	.154
Supervised	L11h10	.658	.699	.04	$-.15 \to -.06$	$-.15$	$-.06$	.233	.252
Supervised†	L9h8	.593	.680	.52	$-.37 \to +.10$	$-.36$	.67	.273	.252
DINOv2 ViT-L	L22h11	.955	.842	.35	$-.32 \to +.15$	$-.31$	.74	—	—

Table 11: **The request mechanism across backbones.** All columns except grouping use the 215 COCO part scenes. Request is the spatial request θ. Best other is the strongest of query, value and final-feature similarity. B − A is attention to person B minus attention to person A, before and after the request swap (Swap) and after the equal-size control (Control). Address R^2 compares requested and actual destinations, and movement R^2 compares predicted and actual address shifts under the request swap, with nothing fitted. Grouping uses the rule of Appendix F on all 1,587 COCO grouping images. †The head chosen most often across 1,000 random development re-splits.

F What does request grouping recover?

F.1 One grouping rule for every representation

Every representation was grouped by the same rule, mean-shift clustering. Mean-shift repeatedly moves each patch vector to the average of the vectors within a fixed radius, and vectors that settle

at the same point form one group. The number of groups is not given in advance: it follows from the radius and the image. The vectors of each image were rescaled so that one radius can serve all images. For each representation, we chose the radius on the 96 grouping-eligible VOC development images, as the one of eleven radii between .15 and 1 with the highest mean FG-ARI plus mIoU (Figure 12). The chosen radii then stay fixed on every COCO image.

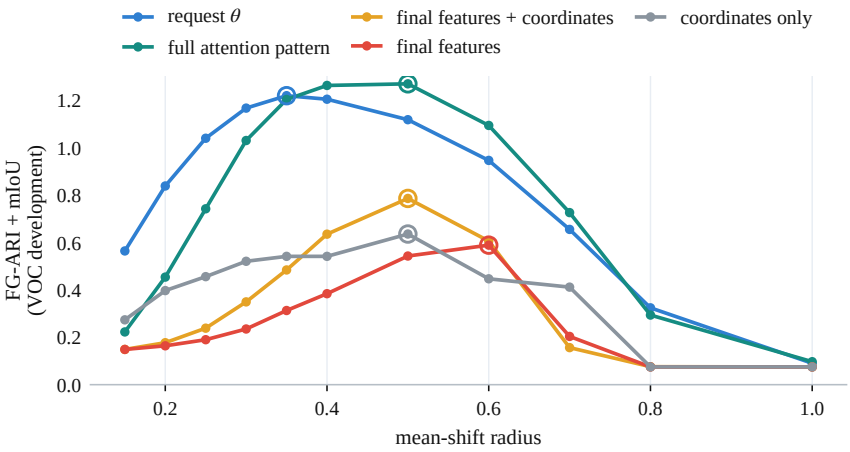

Figure 12: **Choosing the radius.** Mean FG-ARI plus mIoU on the 96 VOC development images for each mean-shift radius, DINOv2. Circles mark the radius kept for each representation.

We grouped the spatial request θ_i and the full attention pattern of the selected head, and the final patch features. Two further baselines verify whether position alone explains request grouping: patch coordinates alone, and final features with patch coordinates appended, whose weight was chosen with the radius. We also evaluated the released COCO model of SPOT (Kakogeorgiou et al., 2024), trained with seven slots on DINO ViT-B/16 features, by assigning each patch to its strongest slot. Evaluated with its own code on all 5,000 COCO validation images, this checkpoint reproduces its published scores (mBO_i .349 and mBO_c .448).

F.2 Instance assignments and object masks

All 1,587 grouping images contain two same-category instances with at least four reliable patches each. The 14×14 grid uses unique instance IDs and 60% patch purity. Crowd-only pixels are ignored and smaller individual masks take precedence in overlaps. The pair test instead excludes ambiguous overlap pixels, so the two tests can select different images. Every method assigns every patch before scoring. FG-ARI measures instance agreement among foreground patches. mIoU scores one-to-one object-mask matches, and mBO_i lets each object choose its best mask. Both mask scores also penalize groups that spill into the background or leave an object incomplete.

Grouping rule	FG-ARI	mIoU	mBO_i
Spatial request θ	.615	**.429**	**.451**
Full attention pattern	**.649**	.403	.436
Final features	.289	.317	.356
Final features + coordinates	.403	.375	.403
Patch coordinates only	.325	.229	.251
SPOT (trained, seven slots)	.356	.288	.350

Table 12: **Routing separates repeated instances better than final features and SPOT, but its masks remain incomplete.** All 1,587 eligible COCO images. The first five rows use the same DINOv2 backbone and mean shift with radii chosen on VOC development. SPOT is the released COCO model, trained on DINO features with seven slots.

Request grouping separates repeated instances far better than final features (.615 versus .289 FG-ARI, higher in 84% of images). Position alone does not explain this: patch coordinates reach .325,

and final features with coordinates added reach .403. SPOT, which learns object slots on COCO, reaches .356, with lower mask scores than the request. Scored instead with the evaluation code of SPOT, the ordering on these images is the same (Table 13). On all 5,000 COCO validation images, SPOT gives better masks, while requests still separate instances better. On the images with repeated objects, SPOT keeps a high class-level score (mBO_c) but a low instance-level score (mBO_i), as when one slot covers several objects of one category. On DINO, the backbone that SPOT itself uses, request grouping reaches .398 (Table 11). The full attention pattern keeps instances apart slightly better (.649), while requests give better object masks. Mask scores stay modest for every representation without training, because background patches also form groups. Figure 14 shows examples with and without the background. Across backbones, request grouping exceeds final-feature grouping in seven of eight (Table 11). The exception is the selected head of the supervised model.

	1,587 images with repeated objects				All 5,000 COCO validation images			
Grouping rule	FG-ARI	mIoU	mBO_i	mBO_c	FG-ARI	mIoU	mBO_i	mBO_c
Spatial request θ	**.548**	**.307**	**.322**	.302	**.410**	.291	.300	.319
Final features	.264	.227	.253	.381	.264	.282	.295	.374
Final features + coordinates	.364	.264	.282	.308	.309	.267	.278	.317
SPOT	.336	.222	.266	**.441**	.368	**.328**	**.349**	**.448**

Table 13: **The comparison with SPOT under its own evaluation code.** Masks at 320 pixels with overlapping pixels ignored, as in SPOT, and our 14×14 groups upsampled to 320 pixels. On repeated objects, requests separate instances better. On general COCO images, SPOT gives better masks.

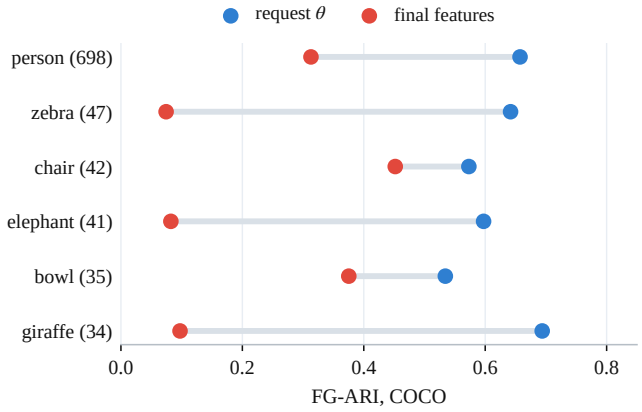

Figure 13: **Request grouping keeps its advantage in common categories.** FG-ARI for the six most frequent categories of the COCO grouping set (number of images in parentheses), DINOv2 with the rule of Table 12.

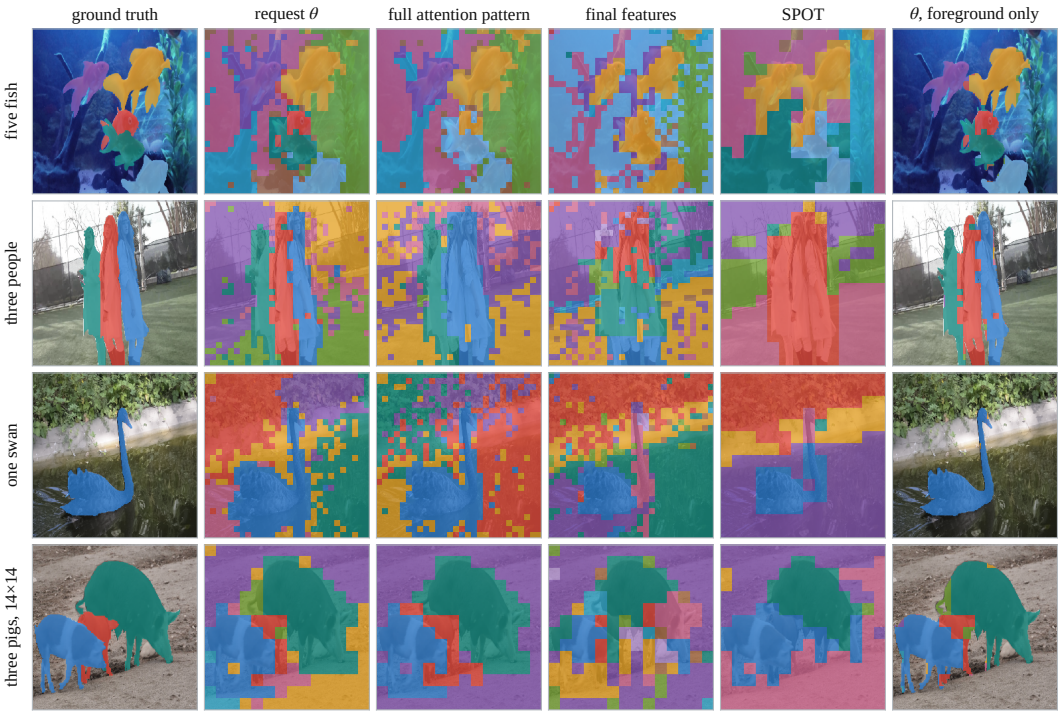

Figure 14: **A useful consequence, and its boundary.** DAVIS frames grouped with the rule of Table 12 on a 28×28 grid (14×14 for the pigs). SPOT is shown with its own 14×14 slots. Colors match corresponding regions across columns. The last column keeps only ground-truth foreground pixels, showing which errors come from including background. This column uses annotations and is not an automatic segmentation result.

F.3 Reading from a region versus finding the patches of an object

Direct attention shows where a query reads. Figure 15 also compares one source patch with every other patch through attention addresses, spatial requests and full attention patterns, all from DINOv2 L10h2. The source is a patch near the center of the largest annotated object. Direct attention is sparse, while the three similarity maps cover much of the source object and also spill into nearby background. Similar requests therefore mark patches that read from the same place, which is not the same as marking a complete object. This is why telling individuals apart and recovering object masks are different tests.

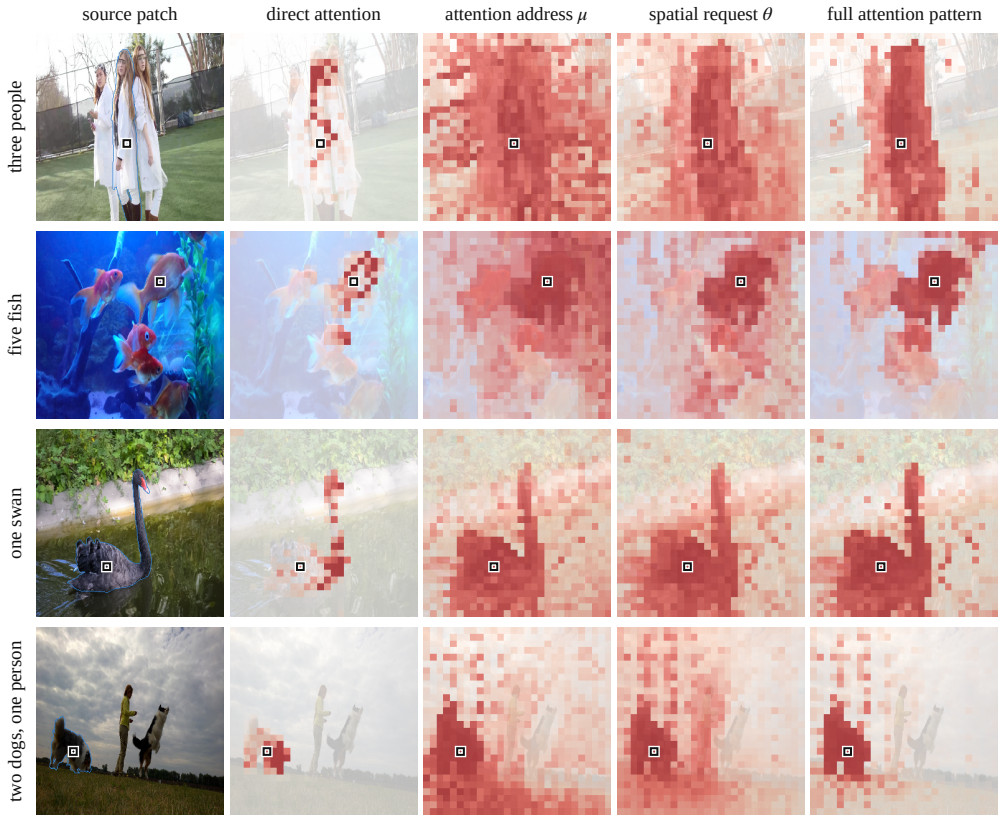

Figure 15: **The attention of one patch and its similarity to other patches.** Black squares mark the source. Only the first column uses its annotated outline. Higher scores appear darker red. For visibility, each panel uses its own minimum-to-99th-percentile contrast range, excluding the source when setting that range. Raw probabilities and similarity scores are not numerically comparable across columns. All maps use the same L10h2 head at 28×28 and show continuous patch values without spatial smoothing.

G FURTHER ANALYSES

G.1 IS THE CONTROLLED TEST ONLY ABOUT INDIVIDUATION?

No. The controlled test removes category and position as cues, but what it finds is not specific to that setting. Keeping L10h2 and its spatial basis fixed, all four readouts also distinguish individuals when the negative object comes from another category (Table 14). Request grouping likewise keeps its advantage across the most common object types (Figure 13). The same routing therefore also separates objects of different categories.

	Different-instance negatives	
Readout	Same category	Different category
Full attention pattern	.862	.915
Attention address μ	.867	.850
Spatial request θ	.866	.845
Requested destination $\widehat{\mu}$	.858	.811

Table 14: **The frozen readouts distinguish individuals across categories too.** Mean same-instance AUC. At each shared offset, we sample equal numbers of same-instance, different-instance/same-category and different-category pairs, at most 12 of each. Each image contributes at least 12 pairs of each kind. All patches are foreground, and only the negative pair type changes between columns.

G.2 HOW DOES THIS WORK DIFFER FROM LI ET AL. (2025)?

Li et al. (2025) studied whether object binding emerges in pretrained ViTs, and decoded same-object membership with a learned probe. We study a more specific question: which native heads keep two objects of the same category apart, and how do they do it (Table 15)? The two tasks differ most in their negative pairs. In our reconstruction of their ADE20K protocol, only 1.0% of pairs come from two instances of one category, so success on that task does not by itself show individuation. The readouts also differ. Their probe is trained on object labels, while our spatial fit uses none and describes a component of the native query that we can edit. On identical objects, their probe loses the separation toward the top of the network, while the late-layer heads of Appendix C.4 still keep identical copies apart.

	Li et al. (2025)	Ours
Question	Does object binding naturally emerge in pretrained ViTs?	How is same-category individuation expressed and used in native attention?
Problem	Binding: do two patches belong to the same object?	Individuation: which of two same-category objects does a patch belong to?
Pairs	All pairs among patches sampled within an image, including background and different categories	Foreground pairs, with negatives from another instance of the same category
Spatial control	No displacement matching	Match distance and direction exactly
Main readout	Learned quadratic probe	Native Q/K relations and the spatial request
Goal	Characterize binding geometry and effects of editing it	Identify specific layers and heads, then explain and edit their native routing

Table 15: **Two complementary questions.**

G.3 WHY DOES THE CONTROLLED DIAGNOSTIC MATTER?

A broad same-object test rewards any readout that knows category or nearness, so it can point to the wrong heads. Figure 16 shows this for one query. With the pairs of Li et al. (2025) on our VOC images, category agreement alone reaches .948 AUC, and the test selects L11h3 key similarity, which responds to both depicted buses. Once background pairs are removed, L10h2 attention rises to the top, but nearness still explains much of the score: nearby patches usually share an object.

L9h7 key similarity shows this cue. It scores well while offsets are unmatched and drops once they are matched. With category and offset both controlled, L10h2 attention is still the selected head, and it reads from the bus of the query. The controls do not create the signal. They remove the easier cues, so that the selected head is the one that separates individuals.

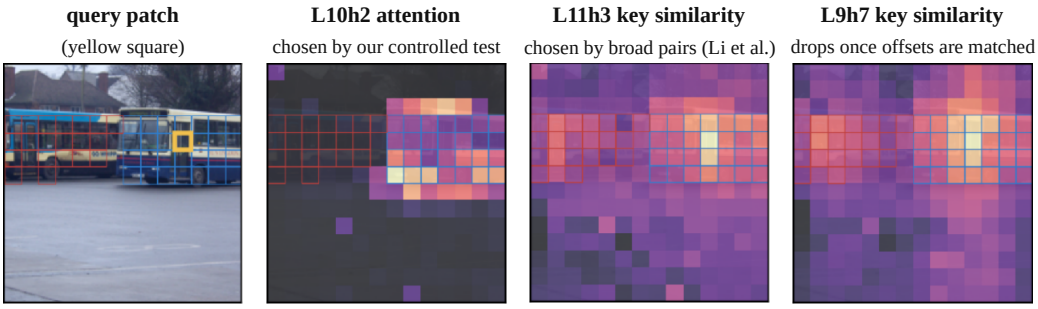

Figure 16: **Broad pair tests highlight different native readouts.** One query on the two-bus scene. L9h7 scores .82 AUC without offset matching and .63 with it. These maps illustrate the comparison, and the population results above establish it.

G.4 Why keep the full request if the destination is two-dimensional?

The requested destination says where a query reads. The full request also says how its preference spreads around that point: two requests can point to the same place while one favors a tight region and the other a wide one. For telling individuals apart, the destination is almost enough: on different parts of one person it reaches .940 AUC, against .942 for the full request. Curvature matters when we are interested in what attention does. With it, the requested destination predicts much better how the address moves under the request swap (Table 7), and the full request groups repeated objects better than its destination alone (.615 versus .580 FG-ARI). Cubic terms add no consistent gain. We keep the quadratic because it describes where and how a query reads, not because identity is five-dimensional.

G.5 Is the selected head stable?

Heads and layers are chosen on the VOC development images, then kept fixed on held-out VOC and COCO. Across 1,000 random re-splits of the 203 VOC images into development and held-out halves, the development rule selects L10h2 attention every time. Across backbones, the rankings of all heads correlate at .93–.98 between the two evaluation sets. In DINOv2, the selected head also has the strongest quadratic fit and request discrimination among all 144 heads (Figure 10). Related instance-sensitive heads occur in the other backbones, although their locations differ. The result concerns a recurring organization of attention, not one universal head index.

H ADDITIONAL VISUAL EXAMPLES

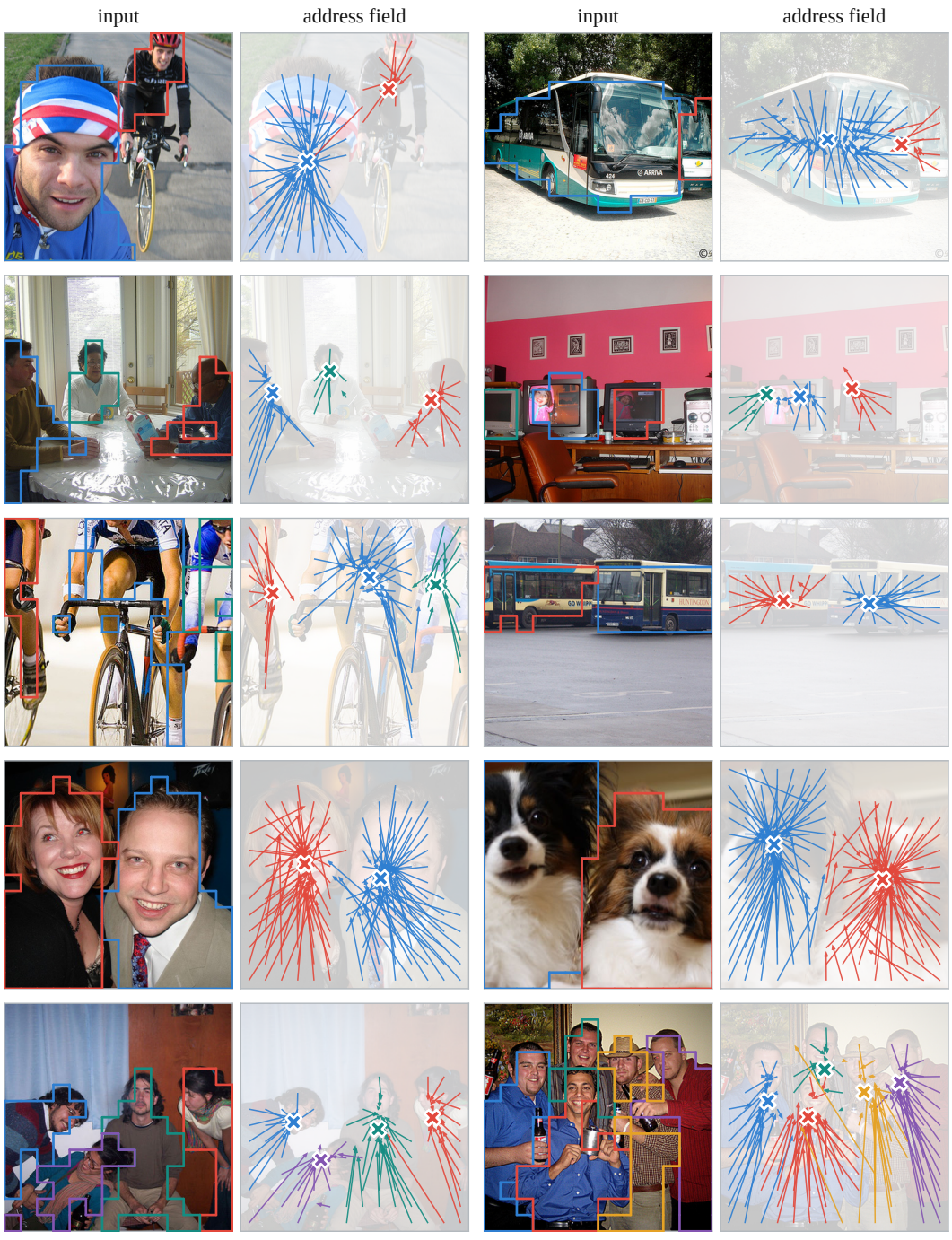

Figure 17: **One frozen DINOv2 head across held-out scenes.** Each pair shows the input and the L10h2 address field. Outlines identify the compared instances. The bottom row contains four and five people. These are illustrative scenes, not a quantitative cohort.

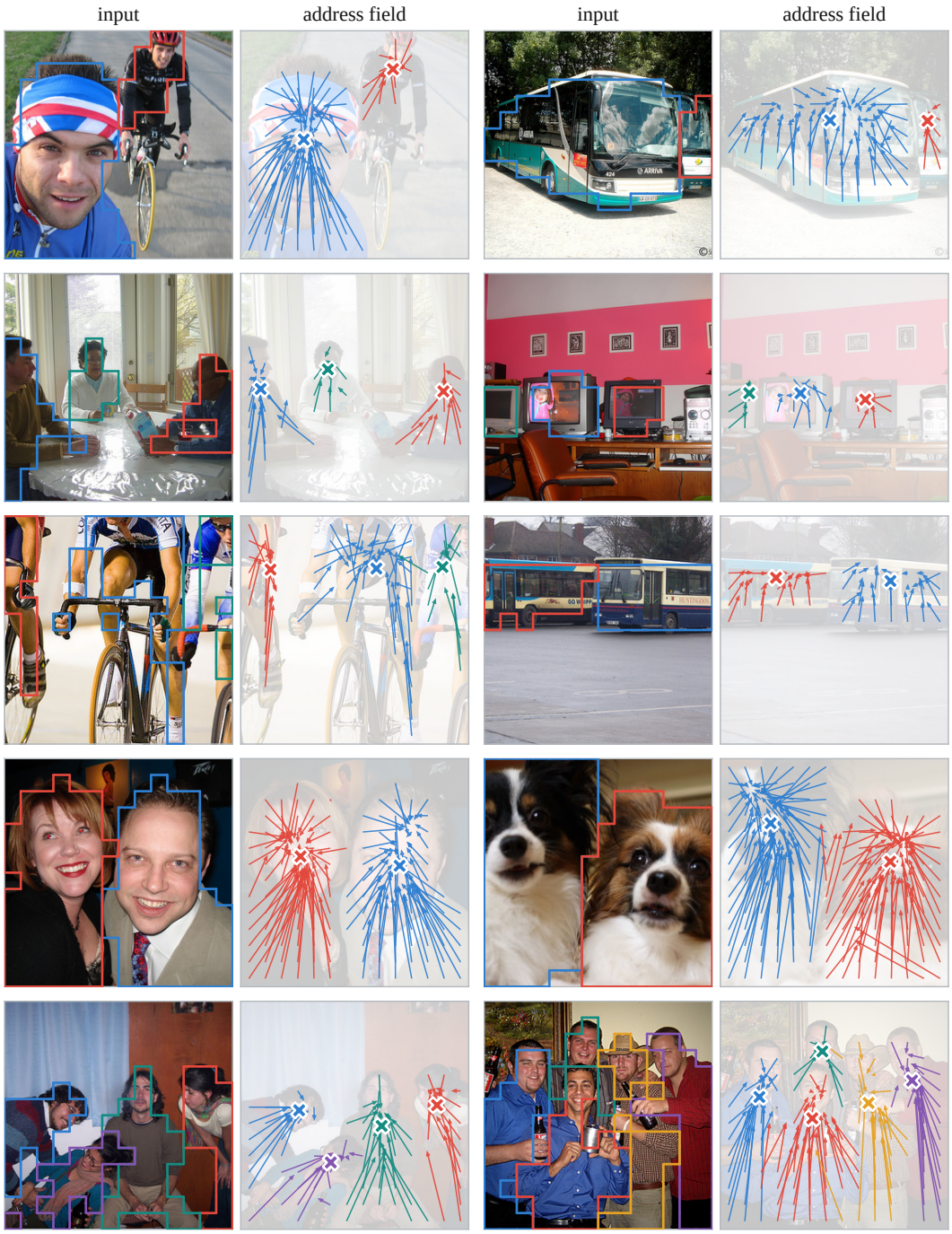

Figure 18: **The same scenes in DINOv3.** Address fields of L11h7. The quantitative comparison uses L10h1, the head selected on the development split. L11h7 is its neighbour, first of the 144 heads on held-out VOC and second on the development split. The bottom row adds four and five people, with each instance shown.

How requests change with depth and neighboring objects. For the layer sequence of Figure 19, a separate spatial basis was fitted at each layer using the same 102 VOC development images, applied on one unchanged scene. It is a qualitative illustration, not another held-out estimate. The nearby-object examples (Fig. 20) show that distinct requests can coexist even when instances touch or overlap. It does not imply that this head separates every touching pair.

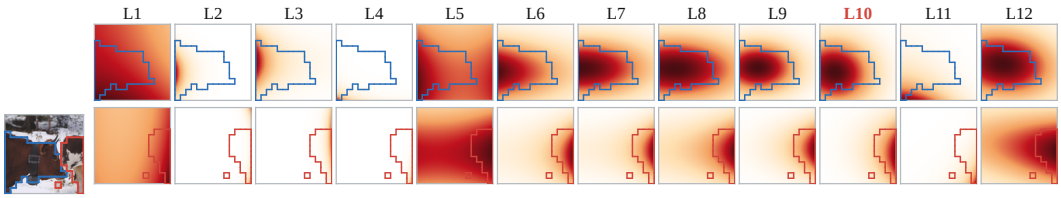

Figure 19: **Spatial requests through depth.** Mean request of each cow at every DINOv2 layer (head 2, with a spatial basis fitted separately per layer). In layers 2–4 both requests sit on the border. From layer 6 to 11 the two cows make distinct requests.

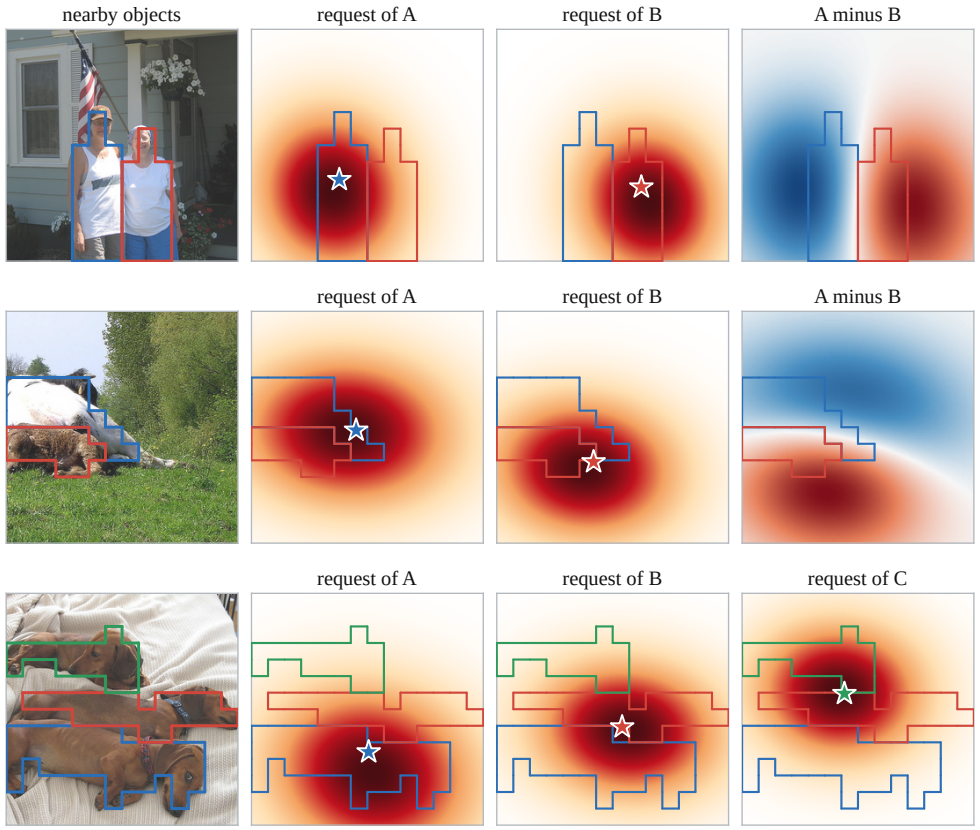

Figure 20: **Nearby instances make distinct requests.** Each row shows nearby instances, their mean requests and requested destinations ($\star$). The final column shows the request difference for two instances, or the request of the third dog.