
Does Object Binding Naturally Emerge in Large Pretrained Vision Transformers?

Yihao Li¹ Saeed Salehi² Lyle Ungar¹ Konrad P. Kording¹

¹University of Pennsylvania ²Machine Learning Group, Technical University of Berlin

liyihao@seas.upenn.edu, ai.neuro.io@gmail.com

ungar@cis.upenn.edu, koerding@gmail.com

Abstract

Object binding, the brain’s ability to bind the many features that collectively represent an object into a coherent whole, is central to human cognition. It groups low-level perceptual features into high-level object representations, stores those objects efficiently and compositionally in memory, and supports human reasoning about individual object instances. While prior work often imposes object-centric attention (e.g., Slot Attention) explicitly to probe these benefits, it remains unclear whether this ability naturally emerges in pre-trained Vision Transformers (ViTs). Intuitively, they could: recognizing which patches belong to the same object should be useful for downstream prediction and thus guide attention. Motivated by the quadratic nature of self-attention, we hypothesize that ViTs represent whether two patches belong to the same object, a property we term *IsSameObject*. We decode *IsSameObject* from patch embeddings across ViT layers using a similarity probe, which reaches over 90% accuracy. Crucially, this object-binding capability emerges reliably in self-supervised ViTs (DINO, MAE, CLIP), but markedly weaker in ImageNet-supervised models, suggesting that binding is not a trivial architectural artifact, but an ability acquired through specific pretraining objectives. We further discover that *IsSameObject* is encoded in a low-dimensional subspace on top of object features, and that this signal actively guides attention. Ablating *IsSameObject* from model activations degrades downstream performance and works against the learning objective, implying that emergent object binding naturally serves the pretraining objective. Our findings challenge the view that ViTs lack object binding and highlight how symbolic knowledge of “which parts belong together” emerges naturally in a connectionist system.¹

1 Introduction

Humans naturally parse scenes into coherent objects [1] (e.g., grouping features such as rounded shape, smooth surface, and muted color into *the mug*) and further ground their identities in context (e.g., recognizing *my coffee mug on the desk* rather than just *a mug*). This is assumed to be made possible by what cognitive scientists call *object binding* [2], the brain’s ability to group an object’s low-level features (color, shape, motion, etc.) into a unified representation. This in turn enables objects to be stored efficiently and compositionally in memory and used as high-level symbols for reasoning. The binding problem is a genuine computational challenge, as evidenced by humans’ limited competence in conjunction-search tasks [3] and clinical dissociations such as Balint’s syndrome, where feature perception remains intact but binding breaks down [4]. If AI systems could replicate the human ability for object binding, that may help them ground symbols for

¹Code available at: <https://github.com/liyihao0302/vit-object-binding>.

perception and exploiting compositionality [5]. The key question is: do current AI systems solve the binding problem?

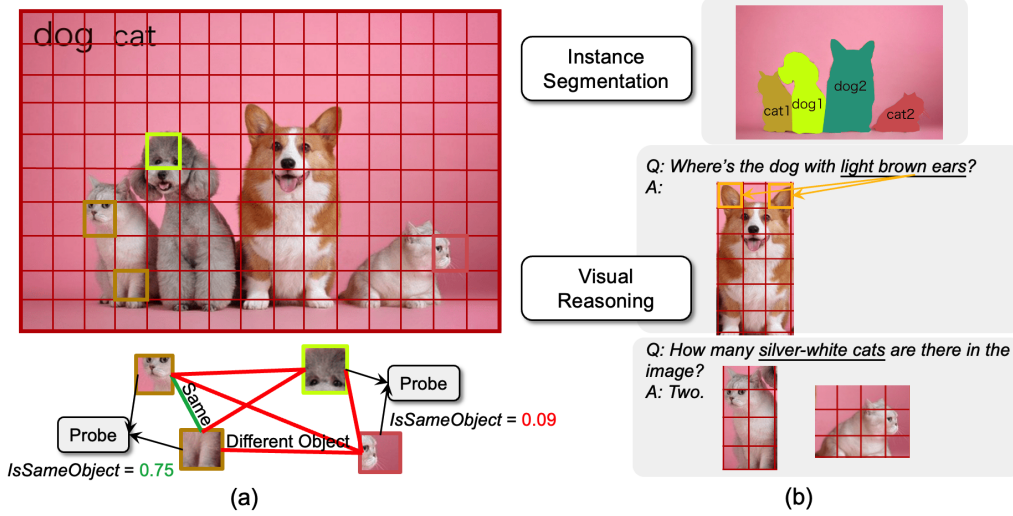


Figure 1: **Assessing object binding in ViTs with *IsSameObject*.** (a) We use a probe to decode *IsSameObject*, with scores near 1 for same-object pairs and near 0 for different-object pairs. (b) Downstream tasks that benefit from strong object binding include instance segmentation and visual reasoning (e.g., locating and counting objects with specific features), where patches triggered by certain features are bound to the rest of their object to allow extraction of the entire object.

Object binding has received little attention in mainstream AI research. Cognition-inspired models [6, 7] build in human-like object-based attention. By contrast, mainstream vision models are assumed to implicitly learn to handle multiple objects from the training set, yet empirical studies show they often "attend" only to the most salient regions and overlook the rest [8]. ViTs compute attention scores from feature-level similarities, which in principle should help with object binding. But experiments show that self-attention behaves more like a similarity-based grouping mechanism than selective attention, which groups image patches by low-level feature similarity (e.g. color or texture), rather than isolating distinct objects [9]. Object-centric methods like Slot Attention [10] fix this by allocating a small set of learnable slots that compete for token features, enforcing binding by design. However, whether AI vision models, especially leading ViTs, can achieve robust object binding without *explicit* mechanisms remains an open question.

Cognitive scientists have questioned whether ViTs can bind objects at all: arguing that they lack mechanisms for dynamically and flexibly grouping features [5]; they lack recurrence necessary for iterative refinement of object representations in the cognitive science field [11, 7]; and as purely connectionist models, they appear incapable of true symbolic processing [8]. However, these architectural limitations do not preclude binding from emerging through learning. If a model encodes whether two patches belong to the same object (*IsSameObject*), this signal can guide attention and improve prediction [12, 13]. Human-labeled data also reflects object-level structure, so ViTs can acquire binding by imitation. This suggests that ViTs may learn to bind objects directly from large-scale training data, without requiring explicit architectural inductive biases.

Here, we ask whether object binding naturally emerges in large, pretrained Vision Transformers, which is a question that matters for both cognitive science and AI. We propose *IsSameObject* (whether two patches belong to the same object) and show that it is reliably decodable (with 90.20% accuracy) using a quadratic similarity probe starting from mid-layers of the transformer layers. This effect is robust across self-supervised models such as DINO, MAE, and CLIP, but largely absent in ImageNet-supervised ViTs, suggesting that binding is an acquired ability rather than a trivial architectural artifact. Across the ViT's layer hierarchy, it progressively encodes *IsSameObject* in a low-dimensional projection-space on top of the features of the object, and it guides self-attention. Ablating *IsSameObject* from model activations hurts downstream performance and works against the pretraining objective.

Our main contributions are as follows: (i) We demonstrate that object binding naturally emerges in large, pretrained Vision Transformers, challenging the cognitive-science assumption that such binding isn’t possible given their architecture. (ii) We show that ViTs encode a low-dimensional signature of *IsSameObject* (whether two patches belong to the same object) on top of their feature representations. (iii) We suggest that learning-objective-based inductive biases can enable object binding, pointing future work toward implicitly learned object-based representations.

2 Related Work

Object Binding in Cognitive Science and Neuroscience. The object binding problem asks how the brain integrates features that are processed across many distinct cortical areas into coherent object representations [14]. The concept of binding² rests on three key hypotheses: First, visual processing is widely understood to be hierarchical, parallel, and distributed across the cortex [16–20]. Second, we perceive the world primarily in terms of objects, rather than as a collection of scattered features [11, 1]. This abstraction is fundamental to both perception and interaction with the world, allowing us to recognize, reason about, and manipulate our environment effectively [21–23]. Third, feature binding requires a mechanism that correctly assigns features, represented in spatially distinct cortical areas, to their corresponding object [15, 24, 2]. This third hypothesis is where the core of the binding problem lies, and it has been a longstanding point of debate among neuroscientists and cognitive scientists [25–27].

Despite their substantial difference, vision transformers (ViTs) share several key computational parallels with the mammalian visual system: they both rely on parallel, distributed and hierarchical processes. More importantly, ViTs do have two of the three architectural and computational elements hypothesized to enable binding in the brain. The explicit position embeddings in ViTs resemble spatial tagging and the spatiotopic organization observed in the ventral stream [28, 26]; and the self-attention mechanism is akin to dynamic tuning and attentional modulation, which are thought to be primary mechanisms for object binding [26, 29, 30] (although attention is believed to be of recurrent nature in the brain [31, 32]). These parallels position ViTs as potential computational models for exploring object binding in both artificial and biological systems.

Object-Centric Learning. Motivated by how humans naturally reason about individual objects, Object-Centric Learning (OCL [10]) aims to represent a scene as a composition of disentangled object representations. While segmentation only partitions an image into object masks, OCL goes further by encoding each object into its own representation [33]. Unsupervised approaches such as MONet [34], IODINE [33], and especially Slot Attention [10] encode scenes into a small, permutation-invariant set of “slots” that are iteratively refined, producing robust object representations on both synthetic [10, 35] and real-world data [36, 37] and enabling compositional generation and manipulation [38–40]. However, slot-based models impose a fixed slot budget and require multiple refinement iterations that slows inference, and because Slot Attention is bolted onto rather than built into the transformer, scaling and training become harder [41]. Other explicit object-centric approaches include Tensor Product Representations [42] and Capsule Networks [43].

Instead of object-centric approaches that explicitly enforce object-level attention, we propose an alternative view that ViTs may already encode implicit object-level structure. Prior work has assumed this and attempted to group patches into objects directly from activations or attention maps ViTs, using methods like clustering [44] or GraphCut [45]. Other studies design self-supervised objectives (e.g., object “movability” [46] or patch-level contrastive learning [47]) and train models to strengthen object-level grouping. In contrast, our study directly validates this assumption by showing that ViT patch embeddings intrinsically encode whether any two patches belong to the same object.

Binding in Transformers. Binding has received growing recognition in transformer-based machine learning research and binding failures are seen as examples of performance breakdowns in modern applications [48–51]. Diffusion models rely on binding attributes to entities, and failures cause attribute leakage (e.g., both a dog and a cat end up wearing sunglasses and a sun-hat) [49, 48]. Vision-language models face similar binding challenges, struggling with differentiating multiple objects with feature conjunctions [51]. Despite these binding failures, transformers still demonstrate some binding

²The term binding was introduced to neuroscience by Christoph von der Malsburg in 1981, inspired by the notion of variable binding in computer science [15].

capability, yet the underlying mechanism is not well understood. Feng and Steinhardt [52], Dai et al. [53] study binding in language models, showing that attributes (e.g., “lives in Shanghai”) are linked to their subjects (e.g., “Alice”) via a low-dimensional *binding-ID* code that is added to the activation and can be edited to swap or redirect relations. Binding mechanisms in vision transformers remain unexplored, and our study aims to fill this gap.

3 Assessing Object Binding in ViTs through *IsSameObject*

3.1 Probing *IsSameObject* representations

Vision Transformers (ViTs) tokenize images by dividing them into a grid of fixed-size patches [54]. Because the token is the minimal representational unit, any grouping of features into objects must arise through relations between tokens, not within them. The only mechanism ViTs have for such cross-token interaction is scaled dot-product attention, where attention scores can be viewed as dynamic edge weights in a graph that route information between tokens [5]. Therefore, if ViTs perform any form of object binding, we expect to observe a pairwise token-level representation that indicates whether two patches belong to the same object, which we term *IsSameObject*.

Since object binding is the ability to group an object’s features together, decoding *IsSameObject* reliably from ViT patch embeddings would provide direct evidence of object binding (and its representation) in the model. We adopt probing, which takes measurements of ViT activations with lightweight classifiers [55], to determine whether *IsSameObject* is encoded or unrecoverable by simple operations.

$$s^{(\ell)} = \text{LayerNorm}(h^{(\ell)} + \text{MultiHeadAttention}(h^{(\ell)})) \quad (1)$$

$$h^{(\ell+1)} = \text{LayerNorm}(s^{(\ell)} + \text{FFN}(s^{(\ell)})) \quad (2)$$

Transformers propagate information across layers according to the above equations 12, where $h^{(\ell)}$ is the residual-stream output (which we’ll call the patch embedding) at layer ℓ .

Formally, we define the *IsSameObject* predicate on a pair of token embeddings $(x_i^{(\ell)}, x_j^{(\ell)})$ at layer ℓ by

$$\text{IsSameObject}(x_i^{(\ell)}, x_j^{(\ell)}) = \phi(x_i^{(\ell)}, x_j^{(\ell)}), \quad \phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1],$$

where ϕ scores the probability that tokens i and j belong to the same object.

We consider the following hypotheses about how *IsSameObject* may be encoded in the model’s activations:

- **It may be *linear*** (recoverable by a weighted sum of features) or **fundamentally *quadratic*** (recoverable only through pairwise feature interactions).
- **It is a *pairwise relationship versus a pointwise mapping*** (i.e. the model first maps each patch to an object identity or class, then compares).
- **The model tells objects apart using only broad *class labels* or object *identities***—i.e., it may solve binding by recognizing “this is a dog vs. this is a chair” without actually representing which specific pixels belong to each object. Such a solution is possible because class labels already provide a coarse notion of object identity without requiring pixel-level grounding.
- **The signal is stored in a few *specialized dimensions versus distributed across many dimensions***. In the former case, binding information would be isolated to a small subset of channels, while in the latter it would be encoded diffusely (e.g., as rotated combinations of features) such that no single dimension carries the signal on its own.

To test these hypotheses, we decode *IsSameObject* using several probe architectures, each defined by a learnable parameter matrix W :

1. **Linear probe:** $\phi_{\text{lin}}(x, y) = Wx + Wy$, where $W \in \mathbb{R}^{1 \times d}$.

2. **Diagonal quadratic probe (specialized dims)** $\phi_{\text{diag}}(x, y) = x^\top \text{diag}(w) y$, with $w \in \mathbb{R}^d$.

3. **Quadratic probe (distributed)** $\phi_{\text{quad}}(x, y) = x^\top W^\top W y$, $W \in \mathbb{R}^{k \times d}$, $k \ll d$.

4. Object-class/identity probes (pointwise) We first map each embedding x, y to class distributions $p = \text{softmax}(W_c x)$, $q = \text{softmax}(W_c y)$ and train W_c with multiclass cross-entropy on object-class labels (and similarly W_N on object-identity labels). The pointwise *IsSameObject* score is then

$$\phi_{\text{class/identity}}(x, y) = p^\top q = \sum_{c=1}^{N_c} p(c) q(c) = \text{softmax}(W_N x)^\top \text{softmax}(W_N y).$$

5. Object-class probes (pairwise) Alternatively, we can treat *IsSameObject* as a binary label and optimize $p^\top q$ directly with binary cross-entropy loss. This version is pairwise because the supervision is applied directly to (x, y) as a pair. The model is thus optimized to encode the *relationship* between two patches, not their individual class memberships.

3.2 *IsSameObject* is best decodable in quadratic form

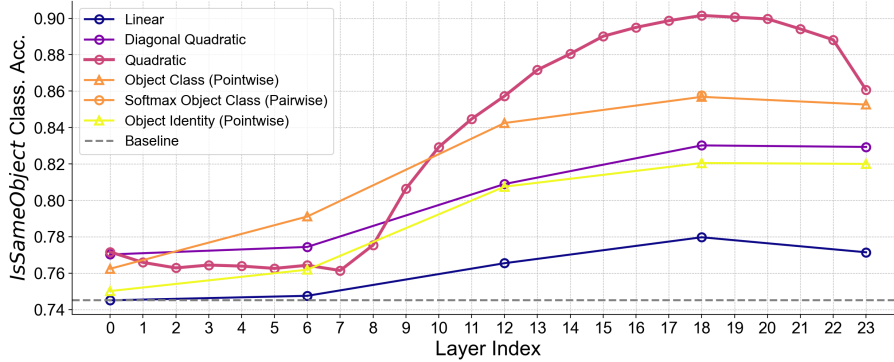


Figure 2: **Quadratic probes excel at decoding the binding signal.** Layer-wise accuracy of the *IsSameObject* probe on DINOv2-Large. The quadratic probe consistently outperforms all other probes from middle layers onward. Results for additional models are shown in Appendix A.2.

We extract DINOv2-Large [13] activations at each layer and train the probes on the ADE20K dataset [56] using cross-entropy loss for all pairwise probes to classify same-object vs. different-object patch pairs. Figure 2 shows probe accuracy across layers. To test our hypotheses about how *IsSameObject* is represented, we compare:

- **Linear vs. quadratic probes:** Quadratic probes (even with the same parameter count as in diagonal quadratic) significantly outperform linear ones, suggesting that *IsSameObject* is a quadratic representation, consistent with the quadratic form used by the self-attention mechanism.
- **Quadratic vs. object-class probes:** Trained object class probes fall short of the quadratic probe directly optimized for distinguishing same and different object instances. This indicates that the model encodes not just whether two patches share an object class but also subtler identity cues—e.g., distinguishing two identical cars of the same make and model.
- **Full vs. diagonal quadratic probes:** The full quadratic probe outperforms its diagonal variant, implying that *IsSameObject* information is more distributed across dimensions rather than restricted to specific channels.
- **Pointwise object class probe vs. pairwise object class probe:** Mapping each patch to an object class via softmax and then comparing classes (pointwise) underperforms directly comparing their embeddings (pairwise), as the pointwise approach loses information when it collapses embeddings into discrete classes.

3.3 Object binding emerges broadly across self-supervised ViTs

We extend our analysis beyond DINOv2 to a broader set of pretrained Vision Transformers, including CLIP, MAE, and fully supervised ViTs. To enable direct comparison, we standardize input patch coverage by resizing all inputs so that each model processes the same spatial patch divisions. Under

this setup, every probe starts from the same trivial baseline of 72.6% accuracy, which corresponds to always predicting “different”, reflecting the class imbalance that most patch pairs do not belong to the same object in the dataset.

Table 1 reports *IsSameObject* decoding accuracy across models. DINO models show the strongest binding signal, with large and giant variants exceeding +16 percentage points over baseline. CLIP and MAE also exhibit clear object-binding ability, though to a lesser degree. In contrast, ImageNet-supervised ViT yields poor object-binding performance, suggesting that binding is an acquired ability under specific pretraining objectives rather than being a universal property of all vision models.

Table 1: **Binding is consistently represented in self-supervised ViTs, but less so in supervised ViTs.** Probe accuracy on *IsSameObject* across pretrained ViTs. Δ is reported in percentage points (pp), and the peak layer index is normalized to $[0, 1]$ within each model.

Model	Highest Accuracy (%)	Δ over Baseline (pp)	Peak Layer (0–1)
DINOv2-Small	86.7	+14.1	1.00
DINOv2-Base	87.5	+14.9	0.82
DINOv2-Large	90.2	+17.6	0.78
DINOv2-Giant	88.8	+16.2	0.77
CLIP (ViT-L)	84.2	+11.6	0.39
MAE (ViT-L)	82.9	+10.3	0.65
Supervised (ViT-L)	76.3	+3.7	0.13

Our findings thus produce a much wider coverage of ViTs and we provide an understanding of potential reasons why binding emerges:

- **DINO.** The contrastive teacher–student loss enforces consistency across augmented views containing the same objects. This objective encourages the model to learn object-level features that persist under augmented views [12].
- **CLIP.** By aligning images with text captions, CLIP effectively assigns each object a symbolic label (e.g., “the red car”), which can act like a pointer that pulls together all patches of that object. This supervision likely encourages patches from the same object to cluster in feature space.
- **MAE.** The masked autoencoder objective requires the model to reconstruct a missing patch from its surroundings. When the masked patch sits between multiple objects, correctly predicting its content forces the model to infer which object it belongs to, thus promoting the grouping of patches from the same object.
- **Supervised ImageNet training.** Because ImageNet supervision labels only the dominant object in each image [57], it rewards class-level categorization but not grouping of patches into object instances, leading to a weaker binding signal.

4 Extracting the Binding Subspace of ViT Representations

4.1 Decomposing *IsSameObject* from features

Following the linear feature hypothesis [58], and similar to [52], we assume that at layer ℓ each token embedding decomposes into a “feature” part and a “binding” part:

$$h^{(\ell)}(x_t) = f^{(\ell)}(x_t, c) + b^{(\ell)}(x_t),$$

where $f^{(\ell)}(x_t, c) \in \mathbb{R}^d$ encodes all attributes of token x_t (texture, shape, etc.) given context $c = \{x_1, \dots, x_T\}$, excluding any information about which other tokens it binds with, and $b^{(\ell)}(x_t) \in \mathbb{R}^d$ encodes the binding information that determines which other tokens belong to the same object (i.e., the *IsSameObject* relation).

Consider two identical patches x_{A_i} and x_{B_i} at corresponding positions of identical objects A and B in the same image, and let their residual be Δ_{AB_i} . It may be tempting to cancel the feature term directly. Indeed, without positional encoding (see proof in Appendix A.4.1), we have $f^{(\ell)}(x_{A_i}) = f^{(\ell)}(x_{B_i})$,

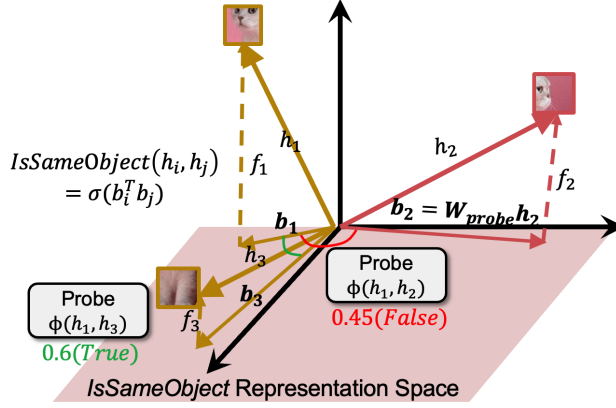


Figure 3: **The geometry of *IsSameObject* representation.** Patch embeddings h_i and h_j are projected onto the *IsSameObject* subspace by W_{probe} , producing binding vectors b_i and b_j , whose similarity is computed by a dot product.

because for identical tokens the positional encoding is the only signal that can differentiate their cross-token interactions.

We can approximate $f^{(\ell)}(x_{A_i}) \approx f^{(\ell)}(x_{B_i})$, since the two patches are visually identical, appear in nearly the same context, and any positional difference can be offloaded into the binding component. This yields:

$$\Delta_{AB_i} = h(x_{A_i}) - h(x_{B_i}) = [f(x_{A_i}) - f(x_{B_i})] + [b(x_{A_i}) - b(x_{B_i})] \approx b(x_{A_i}) - b(x_{B_i}).$$

If Δ_{AB_i} remains roughly consistent across patch pairs with the same index i , then $b(x_{A_i})$ and $b(x_{B_i})$ can form linearly separable clusters, which can thus serve as object identity representations. However, this becomes problematic in natural images, where identical patches are rare.

Instead, we take a *supervised* approach to decoding the binding component. Our quadratic probe serves as a tool for separating binding from feature information within each token (Fig. 3). Conceptually, the quadratic probe can be viewed as projecting an activation h into the *IsSameObject* subspace, yielding $b^{(\ell)}(x) = h^{(\ell)}(x)^\top W$, and then measures the dot-product similarity between two projected vectors. Given that natural image datasets contain numerous objects where b is the primary distinguishing factor, the probe should be optimized to discover a direction that isolates b . With this strategy we can separate the binding signal from the rest of the representation.

The observation that binding vectors remain meaningful under linear combination, and become hard to discriminate when they are close together, is consistent with this interpretation [52]. In later ablation studies, we use our trained quadratic probe via $b^{(\ell)}(x) = h^{(\ell)}(x)^\top W$.

4.2 A Toy Experiment: distinguishing identical objects and similar looking objects

To probe the limits of object binding in ViTs, we construct a test image with two identical red cars, a third red car of a different brand, and a red boat. This setup lets us track *IsSameObject* representations across layers by evaluating three distinctions: different object-class but similar appearance, same class with subtle differences, and exact duplicates. As expected, these distinctions become progressively harder. We chose natural objects rather than abstract shapes because both the ViT and our probe are trained on real-world images, which allows us to analyze binding in a nontrivial setting.

To analyze where binding emerges, we plot the *IsSameObject* scores predicted by our trained quadratic probe (Figure 4). Since the probe performs reliably (see Section 3.2), its outputs approximate the internal *IsSameObject* the model encodes. We observe that, from early to mid-layers, the model increasingly discerns the local object (the one to which each patch belongs). Surprisingly, from mid-layers to later layers, the model shifts toward class-based grouping, increasingly treating all red cars as the same. Binding emerges in the middle of the network and is then progressively lost towards the top.

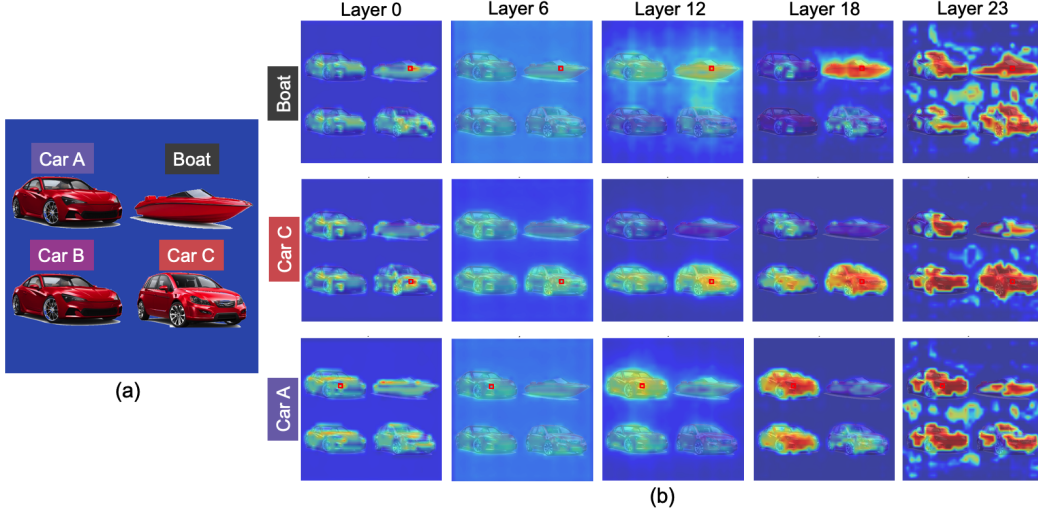


Figure 4: **Layer-wise visualization of *IsSameObject* predictions on the test image.** We used three red cars and one red boat to make binding deliberately difficult. Early layers attend to similar surface features (e.g., the red car or boat hull), mid-layers focus on local objects, and higher layers shift to grouping patches by object class.

The *IsSameObject* representation is low-dimensional. We use four identical red-car images and split each one into patches using exactly the same grid alignment. We perform principal component analysis (PCA) on the residuals sets $\{\Delta_{BA}, \Delta_{CA}, \Delta_{DA}\}$, where $\Delta_{BA} = h_{B_i} - h_{A_i} \approx b_{B_i} - b_{A_i}$ and visualize the first three components (see Figure 5). $\Delta_{BA}, \Delta_{CA}, \Delta_{DA}$ fall into three linearly separable clusters in the first three principal component space, showing that b_B, b_C, b_D form object-level representations. The separation of these clusters in a very small number of principal directions demonstrates that *IsSameObject* lies in a low-dimensional subspace: patches from the same object instance map to closely aligned binding vectors, and different instances are linearly separable with large margins.

Mid-layers capture local objects, and higher layers shift towards grouping patches by object class. A surprising observation is the sudden increase in the cross-object *IsSameObject* score (fig.4) in the mid-layers of the DINOv2 model for instances of the same class (Fig. 4). This is consistent with prior work showing that ViTs represent different types of information at different layers [59]. At the same time, token-position decodability drops in deeper layers (see Appendix A.4.3), suggesting that the model is deliberately discarding positional information. Our interpretation is that the network initially relies on positional cues to support binding, since location is necessary to disambiguate tokens that share similar feature content. However, the DINO training objective enforces position invariance at the output layer [12], which implicitly encourages the network to remove positional signal once it is no longer useful and to repurpose capacity for semantically relevant object structure. Our findings are consistent with experimental evidence from the ventral stream in the brain, showing that while the retinotopic organization of early ventral areas is necessary for perception and binding, global spatial information is instead processed and maintained by the dorsal stream [60–62].

4.3 Attention weights (query-key similarity) correlate with *IsSameObject*

In Section 3.2 we showed that *IsSameObject* is best decoded quadratically. Since self-attention is also a quadratic interaction, binding information in the residual stream at layer ℓ can in principle guide where attention is allocated at layer $\ell + 1$, allowing the model to selectively route attention within the same object to build a coherent object-level representation.

To test this, we compute the Pearson correlation between attention weights and the *IsSameObject* scores (see Fig.6 and Appendix A.5). In mid-level layers, we observe a positive but modest correlation, indicating that the model does make use of the *IsSameObject* signal when allocating attention. The modest strength of the effect is expected, because attention serves many roles beyond binding.

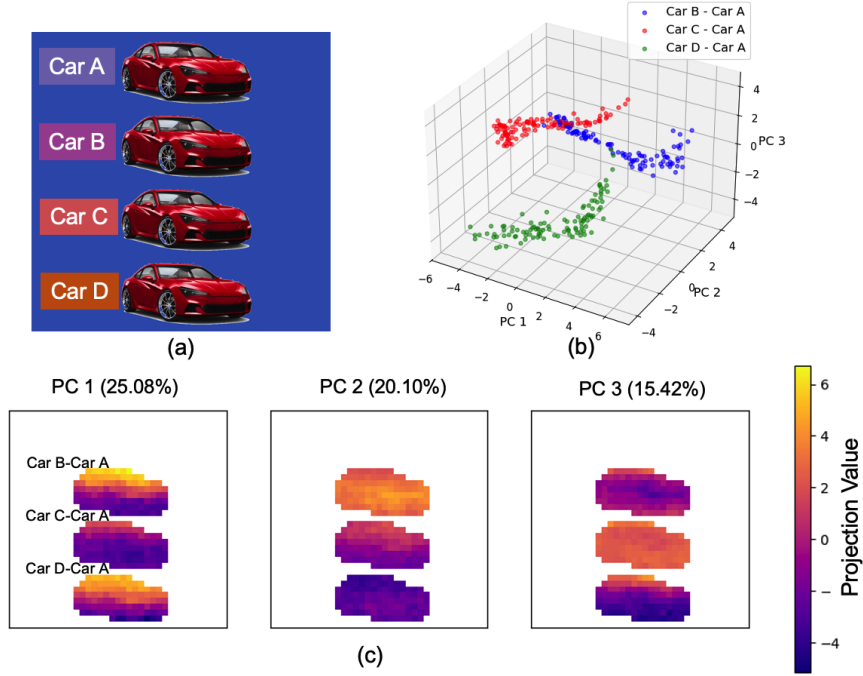


Figure 5: **Identical objects form distinct object-level representations.** The first 3 principal components of the four identical cars, with the three clusters denoting Δ_{BA} , Δ_{CA} , Δ_{DA} . The three linearly separable clusters suggest that identical objects’ binding vectors form distinct object-level representations. The percentage in parentheses indicates the variance explained by that principal component.

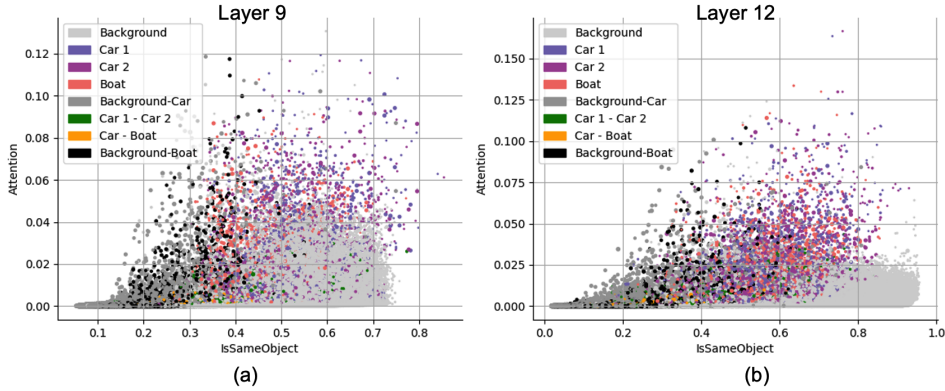


Figure 6: **Attention weights are correlated with *IsSameObject*.** Dot size is proportional to the Euclidean distance between patches. Attention weights correlate with *IsSameObject* in middle layers: (a) Pearson $r = 0.163$, (b) Pearson $r = 0.201$.

4.4 Ablation of *IsSameObject* hurts downstream performance and works against the minimization of the pretraining loss

We conduct ablation studies and evaluate the impact on downstream segmentation performance and pretraining loss. Instead of directly subtracting the *IsSameObject* representation $b(x_i)$ from $h(x_i)$, we use less aggressive approaches:

- **Uninformed Ablation:** Randomly shuffle $b(x_i)$ across patches in the image at a specified ratio

Table 2: **Ablations demonstrate the functional role of *IsSameObject*.** Segmentation accuracy, instance accuracy, and DINO loss on layer 18 under uninformed (random shuffle) and informed (ground-truth injection) ablations. Higher uninformed ratios mean more aggressive shuffling; lower informed α values mean stronger injection of true object labels.

	Uninformed / ratio			Informed / α		
	0	0.5	1	1	0.5	0
Segmentation Acc. (%)	44.14	41.03	39.20	44.14	44.91	43.59
Instance Acc. (%)	35.14	31.39	28.19	35.14	36.37	37.02
DINO Loss	0.6182	0.6591	0.6749	0.6182	—	—

- **Informed Ablation/Injection:** Using ground-truth instance masks, we ablate or inject the true *IsSameObject* signal by linearly combining the mean object direction with each patch’s binding vector b_i : $\tilde{b}_i = (1 - \alpha) \frac{1}{|\mathcal{I}|} \sum_{j \in \mathcal{I}} b_{\text{object},j} + \alpha b_{\text{object},i}$.

We evaluate the semantic and instance segmentation performance with retrained segmentation heads on a subset of ADE20K under these variations. We also evaluate the teacher–student self-distillation loss as employed in DINO (see Appendix A.6 for details).

Results show that uninformed ablation, which randomly shuffles the binding vector, reduces segmentation performance, whereas injecting the mean object direction improves accuracy. Ablating *IsSameObject* with random shuffling leads to a noticeable gradual increase in the DINO loss, suggesting that ablation of *IsSameObject* works against this pretraining loss.

5 Limitations

We assume the trained probe cleanly splits each patch embedding into “feature” and “binding” components, a simplification that would benefit from further empirical exploration. Additionally, while randomizing the *IsSameObject* subspace clearly increases DINO’s pretraining loss, this indirect evidence falls short of demonstrating a causal link to downstream task performance or isolating its precise contribution. Finally, our downstream evaluations focus only on segmentation, leaving open whether these emergent binding signals also benefit other vision tasks such as visual reasoning. More broadly, this paper studies object-identity binding, i.e., whether two tokens belong to the same object; compositional binding, such as reusing object structure across contexts, is not explored here and is left for future work.

6 Conclusion

In this paper, we show that object binding naturally emerges in large, pretrained vision transformers, especially in DINOv2, and this effect is consistent across multiple models. We also show that it is an acquired rather than innate ability through comparisons with supervised models. *IsSameObject*, whether two patches belong to the same object, is reliably decodable and lies in a low-dimensional latent space. Ablating this signal both degrades downstream segmentation and opposes the model’s pretraining loss, suggesting that emergent object binding is a natural solution to the self-supervised objective. Our study bridges what psychologists identify as object binding with emergent behavior in ViTs, challenges the belief that ViTs lack this ability and demonstrates how symbolic processing can arise in connectionist models.

Moving forward, we propose that AI researchers seeking to address binding failures adopt an alternative to explicit binding modules (such as Slot Attention [10]) by enhancing ViTs’ inherent object-binding through tailored training objectives or minimal architectural tweaks. Future work could extend our approach to video models, in which object persistence across time is essential [11, 63]; such models should exhibit similar spatial binding seen in ViTs and additionally develop temporal binding to maintain object identity across frames. Also of interest will be how these object representations formed by binding interact with one another, with one possibility being that they interact with low-dimensional “object files” [64]. Together, these efforts will deepen our understanding of how symbolic processing of objects can emerge in connectionist models.

References

- [1] Brian J Scholl. Objects and attention: The state of the art. *Cognition*, 80(1-2):1–46, 2001.
- [2] Anne Treisman. The binding problem. *Current opinion in neurobiology*, 6(2):171–178, 1996.
- [3] Anne Treisman and Hilary Schmidt. Illusory conjunctions in the perception of objects. *Cognitive psychology*, 14(1):107–141, 1982.
- [4] Lynn Robertson, Anne Treisman, Stacia Friedman-Hill, and Marcia Grabowecky. The interaction of spatial and object pathways: Evidence from balint’s syndrome. *Journal of Cognitive Neuroscience*, 9(3):295–317, 1997.
- [5] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.
- [6] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*, 2018.
- [7] Saeed Salehi, Jordan Lei, Ari S Benjamin, Klaus-Robert Müller, and Konrad P Kording. Modeling attention and binding in the brain through bidirectional recurrent gating. *bioRxiv*, pages 2024–09, 2024.
- [8] Tarun Khajuria, Braian Olmiro Dias, and Jaan Aru. How structured are the representations in transformer-based vision encoders? an analysis of multi-object representations in vision-language models. *arXiv preprint arXiv:2406.09067*, 2024.
- [9] Paria Mehrani and John K Tsotsos. Self-attention in vision transformers performs perceptual grouping, not attention. *Frontiers in Computer Science*, 5:1178450, 2023.
- [10] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.
- [11] Benjamin Peters and Nikolaus Kriegeskorte. Capturing the objects of vision with neural networks. *Nature human behaviour*, 5(9):1127–1144, 2021.
- [12] 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*, pages 9650–9660, 2021.
- [13] 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. *arXiv preprint arXiv:2304.07193*, 2023.
- [14] Christoph Von der Malsburg. The what and why of binding: the modeler’s perspective. *Neuron*, 24(1):95–104, 1999.
- [15] Jerome Feldman. The neural binding problem (s). *Cognitive neurodynamics*, 7:1–11, 2013.
- [16] Semir M Zeki. Functional specialisation in the visual cortex of the rhesus monkey. *Nature*, 274(5670):423–428, 1978.
- [17] Margaret Livingstone and David Hubel. Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, 240(4853):740–749, 1988.
- [18] Mortimer Mishkin, Leslie G Ungerleider, and Kathleen A Macko. Object vision and spatial vision: two cortical pathways. *Trends in neurosciences*, 6:414–417, 1983.
- [19] Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.
- [20] Kalanit Grill-Spector and Kevin S Weiner. The functional architecture of the ventral temporal cortex and its role in categorization. *Nature Reviews Neuroscience*, 15(8):536–548, 2014.

- [21] Gaetano Kanizsa, Paolo Legrenzi, and Paolo Bozzi. Organization in vision: Essays on gestalt perception. (*No Title*), 1979.
- [22] Stephen E Palmer. Hierarchical structure in perceptual representation. *Cognitive psychology*, 9(4):441–474, 1977.
- [23] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.
- [24] Christoph Von Der Malsburg. The correlation theory of brain function. In *Models of neural networks: Temporal aspects of coding and information processing in biological systems*, pages 95–119. Springer, 1994.
- [25] H Steven Scholte and Edward HF de Haan. Beyond binding: from modular to natural vision. *Trends in Cognitive Sciences*, 2025.
- [26] Lynn C Robertson. Binding, spatial attention and perceptual awareness. *Nature Reviews Neuroscience*, 4(2):93–102, 2003.
- [27] Adina L Roskies. The binding problem. *Neuron*, 24(1):7–9, 1999.
- [28] Anne Treisman. Focused attention in the perception and retrieval of multidimensional stimuli. *Perception & Psychophysics*, 22:1–11, 1977.
- [29] John H Reynolds and Robert Desimone. The role of neural mechanisms of attention in solving the binding problem. *Neuron*, 24(1):19–29, 1999.
- [30] Pieter R Roelfsema. Solving the binding problem: Assemblies form when neurons enhance their firing rate—they don’t need to oscillate or synchronize. *Neuron*, 111(7):1003–1019, 2023.
- [31] Ruben S van Bergen and Nikolaus Kriegeskorte. Going in circles is the way forward: the role of recurrence in visual inference. *Current Opinion in Neurobiology*, 65:176–193, 2020.
- [32] Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B Issa, and James J DiCarlo. Evidence that recurrent circuits are critical to the ventral stream’s execution of core object recognition behavior. *Nature neuroscience*, 22(6):974–983, 2019.
- [33] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pages 2424–2433. PMLR, 2019.
- [34] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. URL <https://arxiv.org/abs/1901.11390>.
- [35] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.
- [36] 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. *arXiv preprint arXiv:2209.14860*, 2022.
- [37] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems*, 35: 18181–18196, 2022.
- [38] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.
- [39] Whie Jung, Jaehoon Yoo, Sungjin Ahn, and Seunghoon Hong. Learning to compose: Improving object centric learning by injecting compositionality. *arXiv preprint arXiv:2405.00646*, 2024.

- [40] Jinwoo Kim, Janghyuk Choi, Jaehyun Kang, Changyeon Lee, Ho-Jin Choi, and Seon Joo Kim. Leveraging image augmentation for object manipulation: Towards interpretable controllability in object-centric learning. *arXiv preprint arXiv:2310.08929*, 2023.
- [41] Alexander Rubinstein, Ameya Prabhu, Matthias Bethge, and Seong Joon Oh. Are we done with object-centric learning? *arXiv preprint arXiv:2504.07092*, 2025.
- [42] Wei Yuen Teh, Chern Hong Lim, Mei Kuan Lim, and Ian KT Tan. Towards discrete object representations in vision transformers with tensor products. In *Proceedings of the 2023 7th International Conference on Computer Science and Artificial Intelligence*, pages 190–194, 2023.
- [43] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.
- [44] Jianing Qian, Anastasios Panagopoulos, and Dinesh Jayaraman. Recasting generic pretrained vision transformers as object-centric scene encoders for manipulation policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17544–17552. IEEE, 2024.
- [45] 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, 2023.
- [46] Adam Bielski and Paolo Favaro. Move: Unsupervised movable object segmentation and detection. *Advances in Neural Information Processing Systems*, 35:33371–33386, 2022.
- [47] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Deeply unsupervised patch re-identification for pre-training object detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1348–1361, 2022.
- [48] Maria Mihaela Trusca, Wolf Nuyts, Jonathan Thomm, Robert Honig, Thomas Hofmann, Tinne Tuytelaars, and Marie-Francine Moens. Object-attribute binding in text-to-image generation: Evaluation and control. *arXiv preprint arXiv:2404.13766*, 2024.
- [49] Taihang Hu, Linxuan Li, Joost van de Weijer, Hongcheng Gao, Fahad Shahbaz Khan, Jian Yang, Ming-Ming Cheng, Kai Wang, and Yaxing Wang. Token merging for training-free semantic binding in text-to-image synthesis. *Advances in Neural Information Processing Systems*, 37: 137646–137672, 2024.
- [50] Boshi Wang and Huan Sun. Is the reversal curse a binding problem? uncovering limitations of transformers from a basic generalization failure. *arXiv preprint arXiv:2504.01928*, 2025.
- [51] Declan Campbell, Sunayana Rane, Tyler Giallanza, Camillo Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven Frankland, Tom Griffiths, Jonathan D Cohen, et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.
- [52] Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*, 2023.
- [53] Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in large language models. *arXiv e-prints*, pages arXiv–2409, 2024.
- [54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [55] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

- [56] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.
- [57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [58] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.
- [59] 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.
- [60] Michael J Arcaro, Stephanie A McMains, Benjamin D Singer, and Sabine Kastner. Retinotopic organization of human ventral visual cortex. *Journal of neuroscience*, 29(34):10638–10652, 2009.
- [61] L. G. Ungerleider and L. Pessoa. What and where pathways. *Scholarpedia*, 3(11):5342, 2008. doi: 10.4249/scholarpedia.5342. revision #91940.
- [62] Vladislav Ayzenberg and Marlene Behrmann. The dorsal visual pathway represents object-centered spatial relations for object recognition. *Journal of Neuroscience*, 42(23):4693–4710, 2022.
- [63] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
- [64] 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.
- [65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

A Appendix

A.1 Experimental Setup

Dataset and Preprocessing. Following the DINOv2 standard setup, we use the ADE20K dataset with images resized and cropped to 512×512 pixels, then padded to 518×518 pixels. We employ a patch size of 14×14 , resulting in a total of 1,369 patches per image. All computations are performed using float32 precision on a NVIDIA RTX 4090 GPU.

Training Configuration. We use the Adam optimizer with a learning rate of 0.001 and a step learning rate scheduler with step size of 8 epochs and gamma decay factor of 0.2.

All probes are trained for 16 epochs with a batch size of 256. For each sample image, we randomly select 64 patches and train with supervision.

For **pairwise probes**, we apply supervision over the upper triangular portion of the 64×64 pairwise matrix. The model is optimized using binary cross-entropy loss.

For **pointwise probes**, we consider two tasks:

- Semantic Segmentation: We use standard cross-entropy loss for pixel-level object class classification.
- Instance Segmentation: Following the DETR framework, we employ Hungarian matching for object assignment. The total loss is computed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} + \lambda_{\text{dice}} \mathcal{L}_{\text{dice}} \quad (3)$$

where \mathcal{L}_{cls} is replaced with a weighted binary cross-entropy loss distinguishing between object and no-object classes:

$$\mathcal{L}_{\text{cls}} = -w_{\text{obj}} \log(p_{\text{obj}}) - w_{\text{no-obj}} \log(1 - p_{\text{obj}}) \quad (4)$$

The hyperparameters follow the DETR configuration: `no_object_weight` = 0.1, `mask_weight` (λ_{mask}) = 5.0, `dice_weight` (λ_{dice}) = 5.0, and `num_object_queries` = 100.

A.2 Probe Performance

We train and evaluate quadratic probes on various ViTs including the complete DINOv2 family (small, base, large, giant) and CLIP-L14. All models utilize a patch size of 14×14 pixels. CLIP-L/14 processes 224×224 input images, resulting in 256 patches per image (16×16 grid).

Overall, we observe that DINOv2 models achieve consistent high classification accuracy at 88%, and progress along layers, while CLIP model achieves a lower 84.8% and displays no such growing trend. DINO models seem to have object binding better than CLIP, although it is possible that less fine-grained grid leads to worse *IsSameObject*, since for each patch, the semantics are entangled.

A.3 Cross-Layer Binding

Cross-layer binding may actively happen under these conditions:

- Layer-specific information processing: ViTs represent different types of information at different layers. Although the self-attention implements a "full range" interaction (all tokens to all tokens interaction), the model learns to limit the "interaction" between the correct tokens and integrate features of the object they belong to as we go deeper
- Information retrieval from earlier layers: ViTs may access information from earlier layers when current layers lack necessary binding cues. As shown in Appendix A.4.3 positional information is gradually removed as the network deepens, and other low-level features may also be discarded. This information loss may trigger cross-layer retrieval as a computational strategy, which rather than maintaining all binding-relevant information in every layer, the model efficiently accesses what it needs from earlier representations. This hypothesis requires deeper analysis in future research.

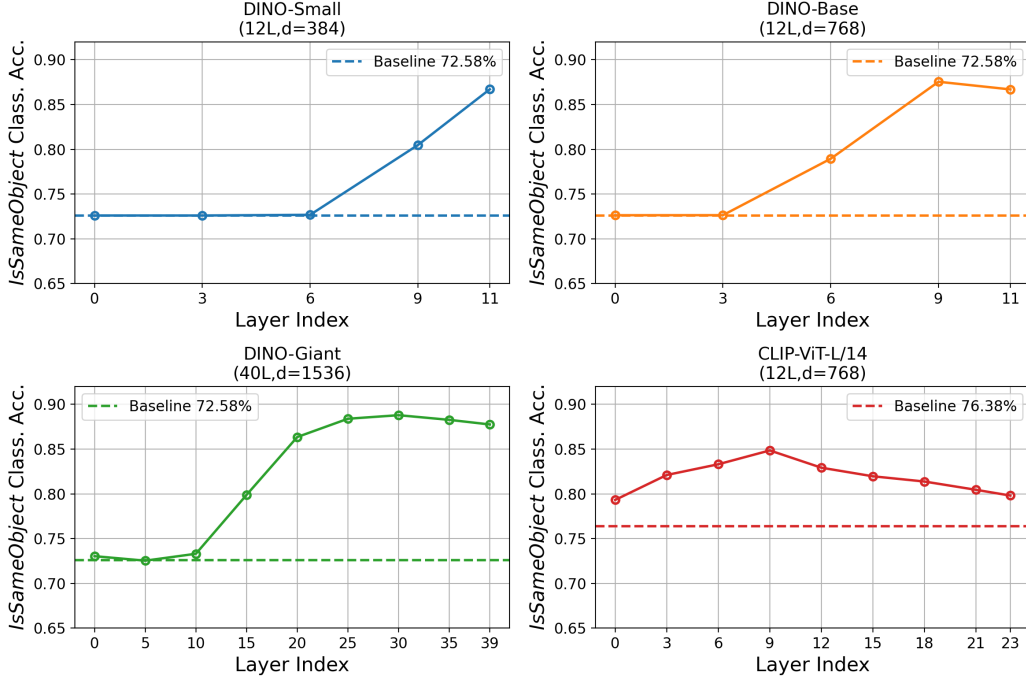


Figure 7: **Layer-wise *IsSameObject* classification accuracy across vision transformer architectures.** DINOv2 models consistently achieve 88% accuracy, demonstrating robust development of *IsSameObject* representations across all model sizes. CLIP-L/14 reaches 85% accuracy and performs closer to the baseline level.

Cross-layer interactions in ViTs can occur via the residual skip connections, so we also train quadratic probes between non-adjacent layers. Specifically, we compute $\phi(x, y) = x^T W_1^T W_2 y$, where W_1 and W_2 are the learned projection matrices from layer 15 and layer 18, respectively. The probe accuracies are: layer 15: 89.0%, layer 18: 90.1%, layer 15-layer18: 83.3%, showing decent cross-layer object binding, and a certain degree of object binding across the system.

A.4 Positional Information

A.4.1 Positional Encoding Distinguishes Identical Objects

A single transformer encoder layer can be viewed as comprising two complementary types of computations: token-wise (i.e., position-wise) operations, which act locally on individual tokens and can be executed in parallel to extract features and short-range interactions; and cross-token operations, implemented through scaled-dot-product attention, which enables long-range interactions by integrating contextual information across all tokens. When inspecting the mathematical formulation, we can also show that for identical tokens (i.e., identical patches), positional encoding is the only information that can guide the cross-token interactions. Here we review the operations for a transformer encoder with a single head from [65] in order. For simplicity, we also assume that query-key vectors have the same dimension as the model (i.e., $d_k = d$). We use blue color for the token-wise operations and red color for cross-token operations. For a sequence of input tokens $\mathbf{t}_i \in \mathbb{R}^k$ where $k = \text{n-channels} \times \text{patch-height} \times \text{patch width}$:

pre-processing:

$$\text{embedding} : \mathbf{e}_i = \mathbf{t}_i \mathbf{W}_E$$

$$\text{adding position embedding} : \mathbf{x}_i = \mathbf{e}_i + \mathbf{p}_i$$

encoder layer:

$$\text{Query-Key-Value} : \mathbf{q}_i = \mathbf{x}_i \mathbf{W}_Q, \mathbf{k}_i = \mathbf{x}_i \mathbf{W}_K, \mathbf{v}_i = \mathbf{x}_i \mathbf{W}_V$$

$$\text{self-attention} : \mathbf{U} = \text{softmax} \left(\frac{\mathbf{Q}(\mathbf{K})^\top}{\sqrt{d}} \right) \mathbf{V}$$

$$\text{projection MLP} : \mathbf{y}_i = \mathbf{u}_i \mathbf{W}_O$$

$$\text{residual connection} : \mathbf{y}_i = \mathbf{x}_i + \mathbf{y}_i$$

$$\text{normalization} : \mathbf{z}_i = \text{LayerNorm}(\mathbf{y}_i)$$

$$\text{feed-forward network} : \mathbf{z}_i = \text{ReLU}(\mathbf{z}_i \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2$$

$$\text{residual connection} : \mathbf{z}_i = \mathbf{z}_i + \mathbf{y}_i$$

where $\mathbf{W}_E \in \mathbb{R}^{k \times d}$ is the embedding layer, $\mathbf{W}_Q \in \mathbb{R}^{d \times d}$, $\mathbf{W}_K \in \mathbb{R}^{d \times d}$, and $\mathbf{W}_V \in \mathbb{R}^{d \times d}$ are the Query, Key, and Value layers, $\mathbf{W}_O \in \mathbb{R}^{d \times d}$ is the linear projection layer, and $\mathbf{W}_1 \in \mathbb{R}^{d \times m}$, $\mathbf{W}_2 \in \mathbb{R}^{m \times d}$ are the feed-forward weights.

Our goal is to show that for two identical tokens (i.e., two patches with identical features), the transformer has to use the position tagging as cue for binding. Since most operations are token-wise (position agnostic), we only need to show the results for the self-attention operation. We will show that if two input tokens are identical with no positional embedding (or with equal positional embedding), then due to the symmetry of the attention mechanism, their output vectors after self-attention will be identical. Formally, if $\mathbf{t}_i = \mathbf{t}_j$ $i \neq j$ and $\mathbf{p}_i = \mathbf{p}_j$ we want to show that $\mathbf{u}_i = \mathbf{u}_j$.

Assuming $\mathbf{t}_i = \mathbf{t}_j$ and $\mathbf{p}_i = \mathbf{p}_j$:

$$\mathbf{x}_i = \mathbf{t}_i \mathbf{W}_E + \mathbf{p}_i = \mathbf{t}_j \mathbf{W}_E + \mathbf{p}_j = \mathbf{x}_j$$

if $\mathbf{x}_i = \mathbf{x}_j$ then:

$$\mathbf{q}_i = \mathbf{q}_j, \mathbf{k}_i = \mathbf{k}_j, \mathbf{v}_i = \mathbf{v}_j$$

Thus the attention score computed by q_i and q_j against all keys would be the same:

$$\mathbf{q}_i^\top \mathbf{k}_n = \mathbf{q}_j^\top \mathbf{k}_n \quad \forall n$$

So the attention weights (after softmax) for rows i and j are the same:

$$a_{i,n} = a_{j,n} \quad \forall n \quad \text{where} : \mathbf{a}_i = \text{softmax} \left(\frac{\mathbf{q}_i \mathbf{K}^\top}{\sqrt{d_k}} \right)$$

And since the values \mathbf{V} are the same across all inputs for the same \mathbf{x}_n , the weighted sum of values will also be identical:

$$\mathbf{u}_i = \sum_{n=1}^N a_{i,n} \mathbf{v}_n = \sum_{n=1}^N a_{j,n} \mathbf{v}_n = \mathbf{u}_j$$

A.4.2 Quantifying the Degree of Distinguishing Identical Objects

Here, we use a simplified form of our proposed toy example containing two identical cars and one red boat.

We quantify the model's ability to distinguish identical objects by examining the kernel density estimation of *IsSameObject* scores between patch pairs from the same object (Car A, Car B, Boat, or Background) across layers in DINOv2-Large in Figure 8.

Ideally, patch pairs from the same object should achieve *IsSameObject* scores approaching 1.0. In early layers, the distributions cluster around 0.5, indicating the model cannot reliably distinguish

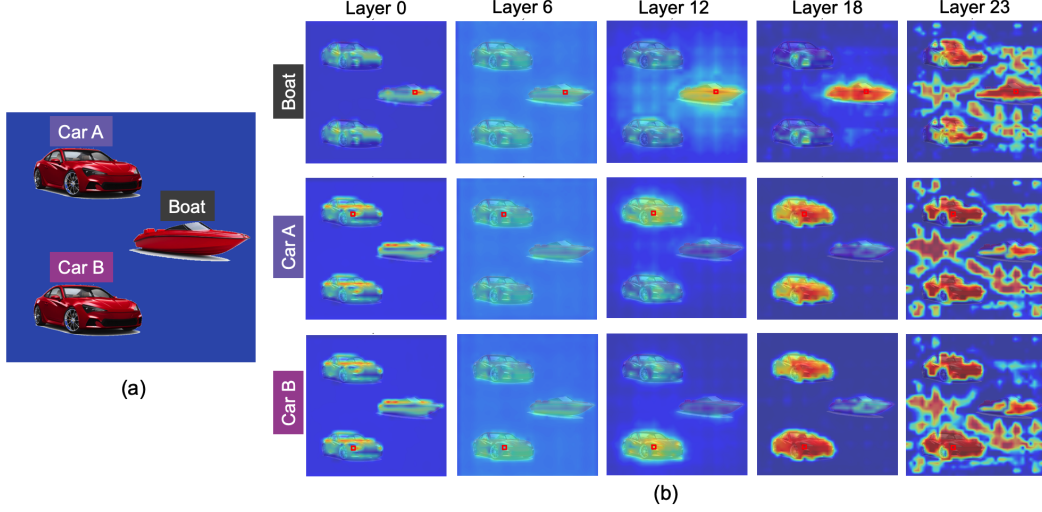


Figure 8: **Layer-wise visualization of *IsSameObject* predictions on the test image** (two identical red cars and one red boat.)

same-object from different-object patch pairs. As processing progresses through later layers, these distributions shift toward 1.0. However, some same-object patch pairs continue to score near 0.0 even in deeper layers, representing indistinguishable token pairs.

We also analyzed patch pairs from different objects in Figure 10, where we expect *IsSameObject* scores to approach 0.0. In layers before 12, the distributions correctly cluster near 0.0, showing the model can distinguish different objects. However, as the model learns to group patches within the same object (as shown in the previous analysis), it simultaneously loses its ability to tell the two identical cars apart. This trade-off is visible in the Car1-Car2 distribution, which gradually shifts upward through the layers and develops a strong peak at 1.0 by the final layer.

A.4.3 Position Information Decay

We hypothesize that the transition from middle layers’ capacity to distinguish identical objects to later layers’ failure stems from the gradual diffusion of precise positional information into more global, semantically-focused representations. To test this hypothesis, we trained linear probes to decode the (x,y) coordinates of each patch from the model’s internal representations (Figure 11). We observe a marked increase in probe RMSE at layer 21, which supports our hypothesis.

A.5 Attention weights (query-key similarity) vs. *IsSameObject*

We investigate the relationship between attention mechanisms and object identity representations by comparing attention weights with *IsSameObject* scores. Attention weights are computed as:

$$\text{Attention}_{ij} = \text{softmax} \left(\frac{Q_i K_j^T}{\sqrt{d_k}} \right) \quad (5)$$

where Q_i and K_j represent the query and key vectors for patches i and j , respectively, and d_k is the key dimension.

We then compute the Pearson correlation between attention weights at layer $\ell + 1$ and *IsSameObject* scores derived from the quadratic probe at layer ℓ :

$$\rho = \text{corr}(\text{Attention}_{ij}^{(\ell+1)}, \text{IsSameObject}_{ij}^{(\ell)}). \quad (6)$$

Using the simplified two-car scenario from Figure 8, we examine how attention weights at layer $\ell + 1$ correlate with *IsSameObject* scores at layer ℓ . In early layers, we observe minimal correlation

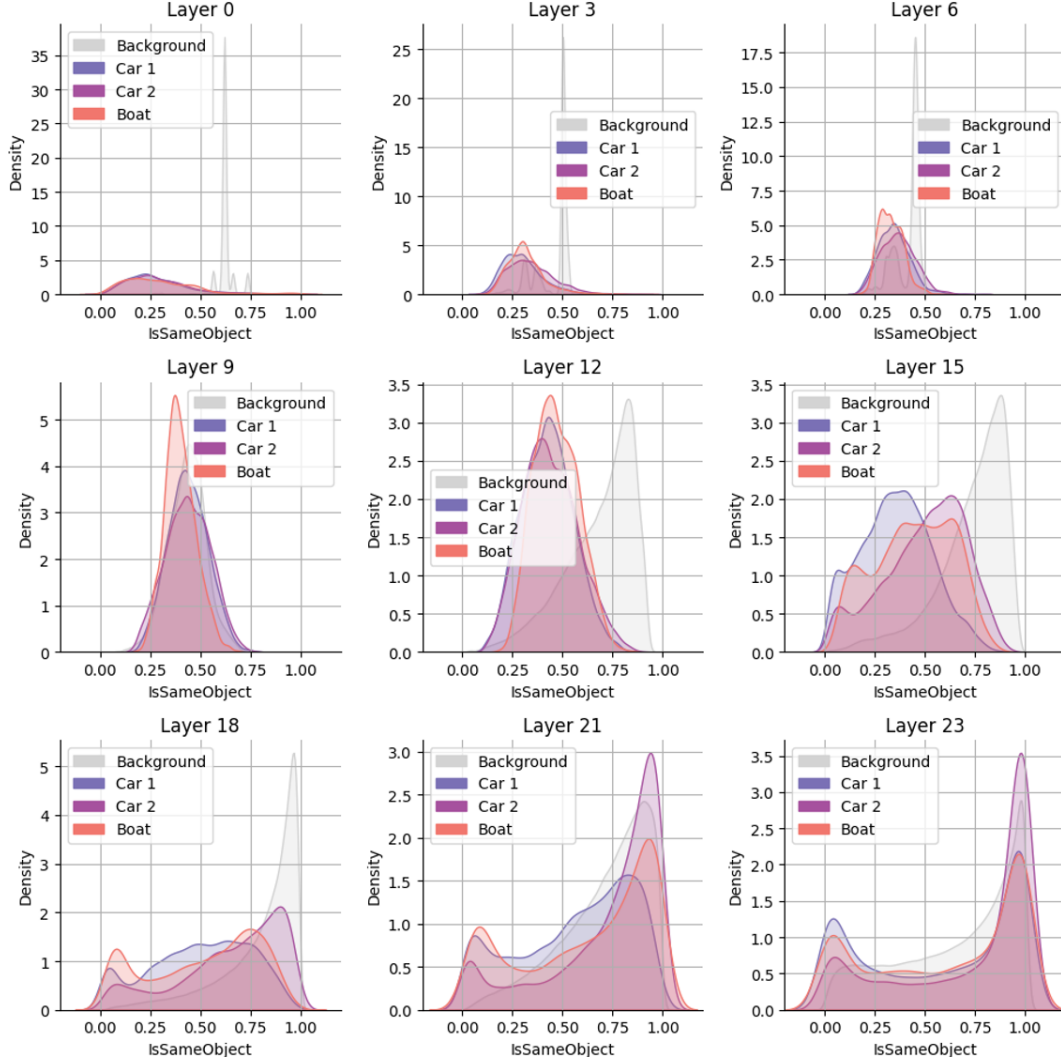


Figure 9: **Binding strengthens with depth for same-object patch pairs.** Kernel density estimation of *IsSameObject* scores for patch pairs within the same object across different layers.

between these two measures, which is likely because *IsSameObject* representation has not yet fully developed.

In deeper layers, certain patch pairs receive high attention weights despite having low *IsSameObject* scores (indicating the model believes they belong to different objects). This phenomenon may be explained by background patches being repurposed for internal computational processes, as identified in prior work on DINO register tokens. Future research could further investigate how these specialized background patches contribute to object representation and their role in maintaining distinct “object files”.

The Pearson correlations in Fig. 6 are statistically significant ($p < 0.001$ under permutation test).

A.6 Implementation of Ablation Studies.

We conduct ablation experiments at layer 18 of DINOv2-Large, where *IsSameObject* representation achieves the best decodability. We apply both uninformed and informed ablation methods as described in Section 3.3.

Segmentation Evaluation. For both semantic and instance segmentation tasks, we retrain linear segmentation heads with ablated representations. The uninformed ablation randomly permutes

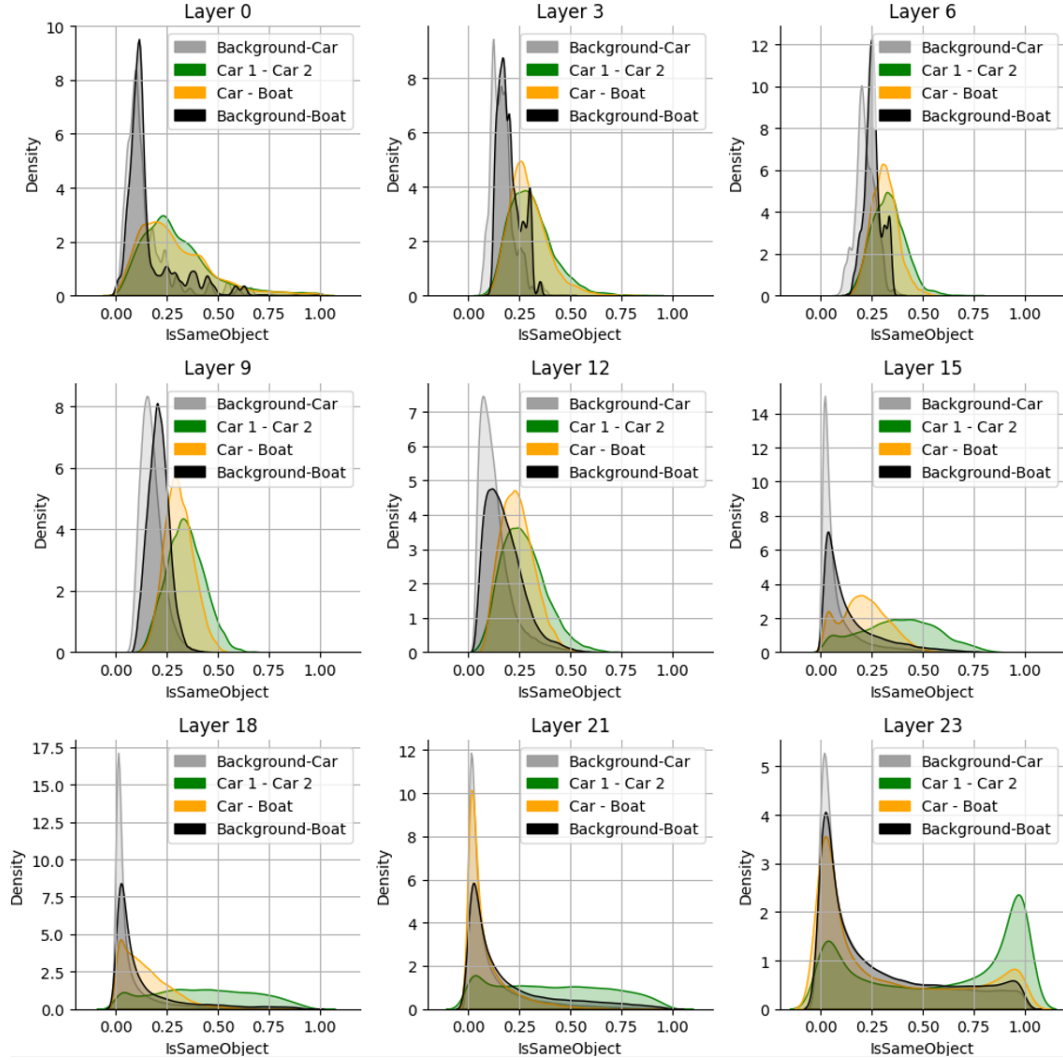


Figure 10: **Identical objects collapse in representation at deeper layers.** Kernel density estimation of *IsSameObject* scores for patch pairs from different objects across layers.

binding vectors across patches, while the informed ablation injects object-averaged binding vectors using ground-truth masks. These linear heads use identical configurations to the pointwise probes described in Section A.1, which are effectively pointwise probes applied to the final transformer layer.

DINO Loss Evaluation. To assess the impact on the pretraining objective, we evaluate DINO loss using the pretrained model as both student and teacher networks. For computational simplicity, we exclude the iBOT and KoLeo loss components from this analysis. Note that informed ablation cannot be evaluated under DINO loss, as the use of local crops alters the patch divisions, making object-averaged binding vectors undefined for the cropped regions.

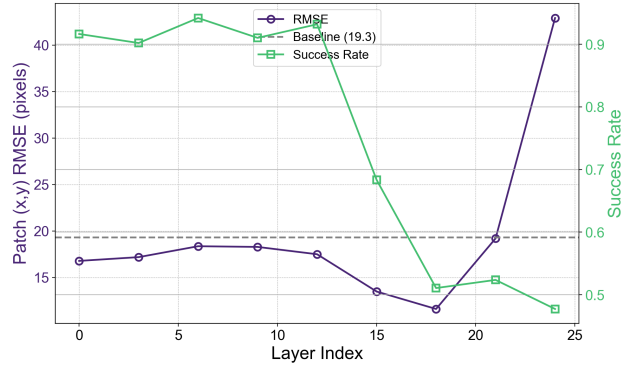


Figure 11: **Positional information decays in later layers.** Layer-wise decoding performance for patch (x,y) coordinates, compared with the success rate of distinguishing patches from Car A and Car B.

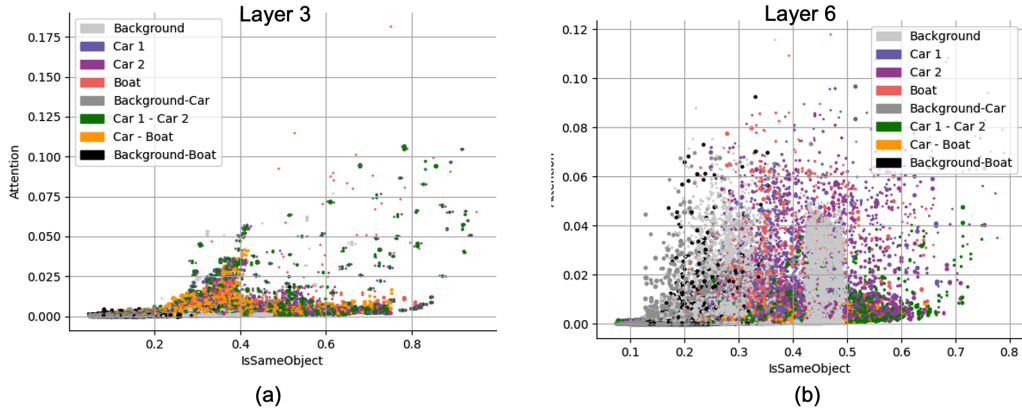


Figure 12: **Attention vs. *IsSameObject* in early layers.** Scatter plots comparing attention weights to *IsSameObject* scores for patch pairs; correlation is still weak, indicating that binding has not yet developed sufficiently to influence attention.

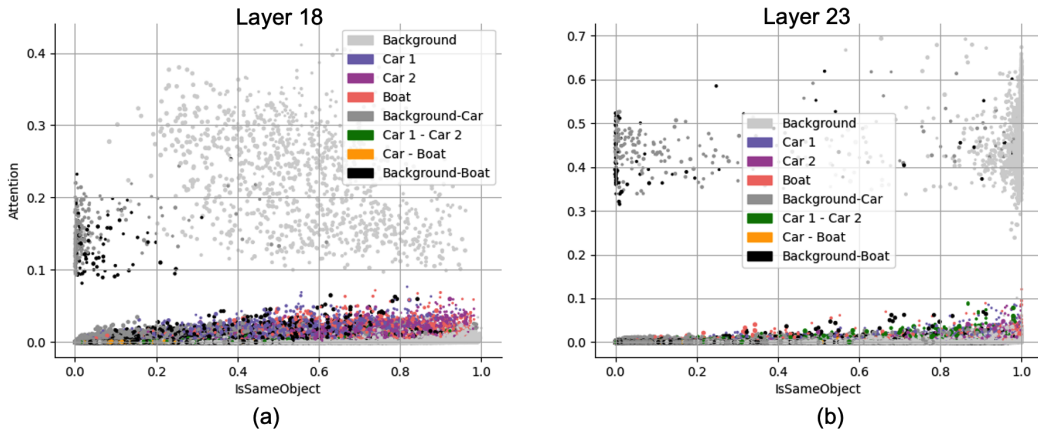


Figure 13: **Attention vs. *IsSameObject* in later layers.** Attention is sometimes allocated to low-*IsSameObject* background tokens, suggesting these tokens might be repurposed for internal computation.

NeurIPS Paper Checklist

1. Claims

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

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

Justification: The abstract and introduction explicitly state the core contributions (i) demonstrating that object binding emerges naturally in large pretrained ViTs, (ii) identifying its low-dimensional, and (iii) showing that this binding signal benefits both downstream tasks and the pretraining objective.

Guidelines:

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

2. Limitations

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

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

Justification: We discuss the limitations of the work in Section 5.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [No]

Justification: Key conditions in Section 4.1 for the linear decomposition hypothesis are described informally without rigorous specification.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: The paper gives a complete description of the experimental setup in Section 4 and implementation details in the Appendix.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: We have released our code and scripts in a public GitHub repository.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: The paper gives a complete description of the experimental setup in Section 4 and implementation details in the Appendix.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [Yes]

Justification: We report Pearson r values alongside their p -values ($p < 0.001$) to convey statistical significance in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: Detailed compute specifications are provided in the Appendix.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: The study uses only publicly available datasets (ADE20K, ImageNet), involves no human subjects or sensitive personal data.

Guidelines:

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

10. Broader impacts

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

Answer: [No]

Justification: Our work focuses on internal representational analysis of pretrained ViTs and does not have direct societal impacts to discuss.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. Safeguards

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

Answer: [NA]

Justification: The paper does not involve releasing new models or high-risk datasets, so no additional safeguards are required.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We credit the DINO framework to Caron et al. (MIT License), ADE20K dataset under its CC-BY-SA License, and the ViT models via Hugging Face (Apache 2.0).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: All newly introduced assets are documented in the Appendix.

Guidelines:

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

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

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

Answer: [NA]

Justification: The study uses only pretrained models and public image datasets, with no human participants or crowdsourced annotation tasks involved.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper do not involve human subjects or participant data.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: Any use of LLMs was limited to writing, editing, and formatting the manuscript; they did not contribute to the core methodology or scientific results.

Guidelines:

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