Deep ViT Features as Dense Visual Descriptors

Shir Amir* Yossi Gandelsman‡ Shai Bagon† Tali Dekel*

*Dept. of Computer Science and Applied Math, The Weizmann Institute of Science

‡Berkeley Artificial Intelligence Research (BAIR)

† Weizmann Artificial Intelligence Center (WAIC)

Project Website: dino-vit-features.github.io

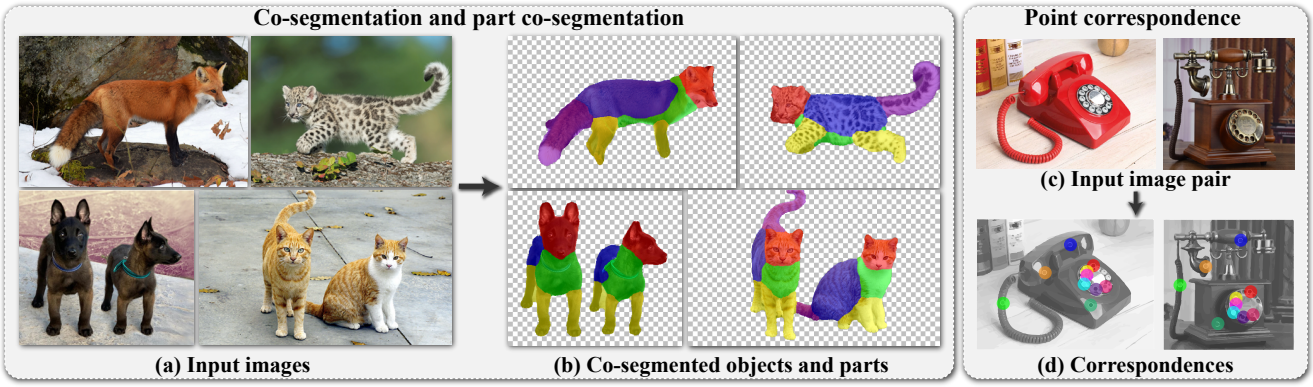


Figure 1: **Deep ViT Features applied to vision tasks.** We demonstrate the effectiveness of deep features extracted from a self-supervised, pre-trained ViT model (DINO-ViT) as *dense patch descriptors* via real-world vision tasks: (a-b) **co-segmentation & part co-segmentation:** given a set of input images (e.g., 4 input images), we automatically co-segment semantically common foreground objects (e.g., animals), and then further partition them into common parts; (c-d) **point correspondence:** given a pair of input images, we automatically extract a sparse set of corresponding points. We tackle these tasks by applying only lightweight, simple methodologies such as clustering or binning, to deep ViT features. These results demonstrate our key observation that the features capture *powerful semantic information, with high spatial granularity, across different objects*, under significant variations in appearance and pose.

Abstract

We leverage Deep Features extracted from a pre-trained Vision Transformer (ViT) as dense visual descriptors. We demonstrate that such features, when extracted from a self-supervised ViT model (DINO-ViT), exhibit several striking properties: (i) the features encode powerful high level information at high spatial resolution–i.e., capture semantic object parts at fine spatial granularity, and (ii) the encoded semantic information is shared across related, yet different object categories (i.e. super-categories). These properties allow us to design powerful dense ViT Descriptors that facilitate a variety of applications, including co-segmentation, part co-segmentation and correspondences – all achieved by applying lightweight methodologies to deep ViT features (e.g., binning / clustering). Our methods, extensively evaluated qualitatively and quantitatively, achieve state-of-the-art part segmentation results, and competitive results with recent supervised methods trained specifically for co-segmentation and correspondences. We take these applications further to the realm of inter-class object co-segmentation and part co-segmentation – demonstrating how objects from related categories can be commonly segmented into semantic parts, under significant pose and appearance changes.

1. Introduction

"Deep Features" – features extracted from the activation of layers in a pre-trained neural network – have been extensively used as visual descriptors in a variety of visual tasks, yet have been mostly explored for CNN-based models. For example, deep features extracted from CNN models that were pre-trained for visual classification (e.g., VGG) have been utilized in numerous visual tasks including image generation and manipulation, correspondences, tracking and as a general perceptual quality measurement.

Recently, Vision Transformers (ViT) [13] have emerged as a powerful alternative architecture to CNNs. ViT-based models achieve impressive results in a variety of visual tasks, while demonstrating better robustness to occlusions, adversarial attacks and have less texture bias compared to CNN-based models [35]. This raises the following questions: Do these properties reflect on the internal representation learned by ViTs? Should we consider Deep ViT Features as an alternative to deep CNN features? Aiming to answer these questions, we explore the use of deep ViT features as general dense visual descriptors: we empirically study their unique properties, and demonstrate their power through a number of real-world visual tasks.

In particular, we focus on two pre-trained ViT models: a

supervised ViT, trained for image classification [13], and a self-supervised ViT (DINO-ViT), trained using a self-distillation approach [3]. We dive into the self-attention modules learned by these models across layers, and empirically demonstrate that DINO-ViT features: (i) encode powerful high level information at high spatial resolution, i.e., capture semantic object parts at fine spatial granularity, and (ii) the encoded semantic information is shared across related, yet different object categories (i.e. super-categories). We demonstrate that these properties are not only due to the ViT architecture but also significantly influenced by the training supervision.

Equipped with these observations, we unlock the effectiveness of DINO-ViT features for a number of vision tasks: co-segmentation, part co-segmentation, and point correspondences. For all these tasks, we develop simple, lightweight methodologies that are applied to deep ViT features (e.g., clustering/binning), and do not require further training. In contrast to most co-segmentation and part co-segmentation methods that require a dataset of annotated images from a specific domain/class for training (e.g., human faces, birds), our framework can be readily applied to an arbitrary number of input images, ranging from as little as a pair of images to a collection containing thousands of images. Our framework can be applied to inter-class co-segmentation and part co-segmentation tasks (across different categories that are semantically related). To our knowledge, we are the first to show results of part co-segmentation in such challenging cases (Fig. 1(a-b)). We thoroughly evaluate our performance qualitatively and quantitatively and demonstrate state-of-the-art performance w.r.t. existing part co-segmentation methods on known benchmarks, and superior results for point correspondences w.r.t. CNN-based feature correspondence methods under significant pose and appearance variations.

To conclude, our key contribution is twofold: (i) providing new empirical observations on the internal features learned by ViT under different supervisions (ii) harnessing these features and their unique properties for several vision tasks; we demonstrate new capabilities in representing objects at a fine spatial granularity, across super-categories, in challenging real-world scenarios.

2. Related Work

CNN-based Deep Features. Features of pre-trained CNNs are a cornerstone for a plethora of vision tasks from object detection and segmentation [18, 6], to image generation [43]. These representations were shown to align well with human perception [16, 23, 53, 33] and to encode a wide range of visual information - from low level features (e.g. edges and color) to high level semantic features (e.g. object parts) [37, 5]. Nevertheless, they exhibit a strong bias towards texture [17], and lack positional information due to their shift equivariance [52]. Moreover, their restricted receptive field [32] makes them capture mostly local information and ignore long-range dependencies [48]. Here, we study as an alternative, deep features of a less restrictive architecture–the Vision Transformer.

Vision Transformers (ViTs). Recently, Vision Transformers (ViT) [13] emerged as a powerful alternative architecture to CNNs. ViT-based models achieve impressive results in a variety of visual tasks [13, 8, 2], while demonstrating better robustness to occlusions and adversarial attacks and have less texture bias compared to CNN-based models [35]. Caron et al. [3] presented DINO-ViT – a ViT model trained without labels, using a self-distillation approach. The effectiveness of DINO-ViT representations was exhibited through impressive results on several downstream tasks, including image retrieval, object segmentation, and copy detection.

Nevertheless, previous works [3, 45] only scratched the surface of utilizing the full potential of deep ViT features – they only considered output features extracted from the last layer, and their use as global or spatially coarse representations. We take the use of deep ViT features a step forward by empirically examining the continuum of Deep ViT features across layers, and their use as *local, dense visual descriptors*.

Concurrently, [39, 11, 35] study theoretical aspects of the underlying machinery, aiming to analyze how ViTs process visual data compared to CNN-based models. Our work aims to bridge the gap between better understanding Deep ViT representations and their use in real-world vision tasks.

Co-segmentation. Co-segmentation aims to jointly segment objects common to all images in a given set. Several unsupervised methods used hand-crafted descriptors [15, 41, 42] for this task. Later, CNN-based methods applied supervised training [29] or fine-tuning [51, 27, 28] on *intra-class* co-segmentation datasets. The supervised methods obtain superior performance, yet their notion of "commonality" is restricted by their training data. Thus, they struggle generalizing to *inter-class* scenarios. We, however, show an unsupervised approach that is competitive to supervised methods for intra-class co-segmentation *and* outperforms them in the inter-class setting.

Part Co-segmentation. Given a set of images with similar objects, the task is to discover common object parts across the similar objects. Recent methods [20, 30] train a CNN encoder-decoder in a self-supervised manner to solve this task, while [10] applies matrix factorization on pre-trained deep CNN features for this task. In contrast, we utilize a pre-trained self-supervised ViT to solve this task, and achieve superior performance to the methods above. We do this by extending our co-segmentation approach to apply part co-segmentation. In addition, to the best of our knowledge, we are the first to perform part co-segmentation in an inter-class scenario.

Semantic Correspondences. Given a pair of images, the task is to find semantically corresponding points between them. [1] propose a sparse correspondence method for inter-class scenarios; leveraging pre-trained CNN features. Recent works employ transformers for dense correspon-

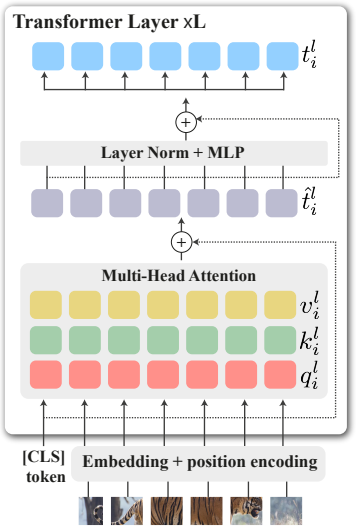


Figure 2: **ViT model:** An image is split into n non-overlapping patches and receives a [CLS] token. They are positionally embedded and passed through transformer layers. Each patch is directly associated with keys, queries, values and tokens in corresponding locations across the transformer layers.

dence in intra-class pairs [8, 46, 22]. However, those methods fail to find meaningful correspondences under significant pose, scale and appearance changes. We show utilizing ViT features can result in higher robustness in these cases.

3. ViT Features as Local Patch Descriptors

We explore ViT Features as *local patch descriptors*. In a ViT architecture, an image is split into n non-overlapping patches $\{p_i\}_{i\in 1..n}$ which are processed into *spatial tokens* by linearly projecting each patch to a d-dimensional space, and adding learned positional embeddings. An additional [CLS] token is inserted to capture global image properties.

The set of tokens are then passed through L Transformer Encoder layers, each consists of normalization layers (LN), Multihead Self-Attention (MSA) modules, and MLP blocks (with skip connections), namely:

$$\hat{T}^l = \mathsf{MSA}(\mathsf{LN}(T^{l-1})) + T^{l-1}$$
$$T^l = \mathsf{MLP}(\mathsf{LN}(\hat{T}^l)) + \hat{T}^l$$

where $T^l = \left[t_0^l, \dots t_n^l\right]$ are the output tokens for layer l.

In each MSA block, the (normalized) tokens are linearly projected into queries, keys and values:

$$q_i^l = W_q^l \cdot t_i^{l-1}, \quad k_i^l = W_k^l \cdot t_i^{l-1}, \quad v_i^l = W_v^l \cdot t_i^{l-1}$$

which are then fused using multihead self-attention. Figure 2 illustrates this process, for full details see [13].

Apart from the initial sampling of the image patches, ViT architecture has no additional spatial sampling, hence, each image patch p_i is directly associated with its query, key, value or token at each layer l.

We next focus our analysis on using the *keys* as 'ViT features'. We justify this choice via ablation in Sec. 4.

3.1. Properties of ViT's Features

We focus on two pre-trained ViT models, both have identical architecture, and both trained on ImageNet data, but differ in the training supervision: a *supervised ViT*, trained

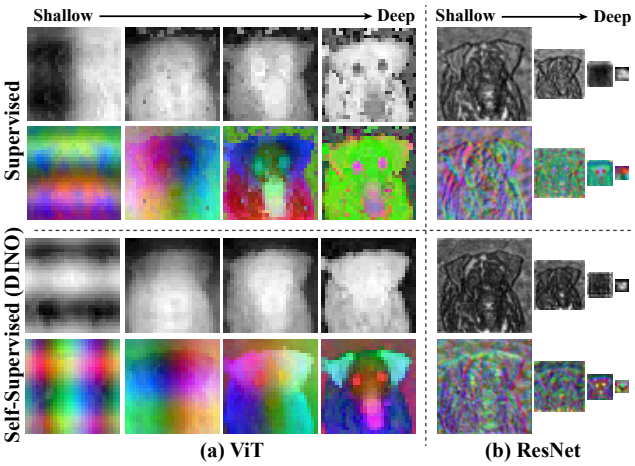
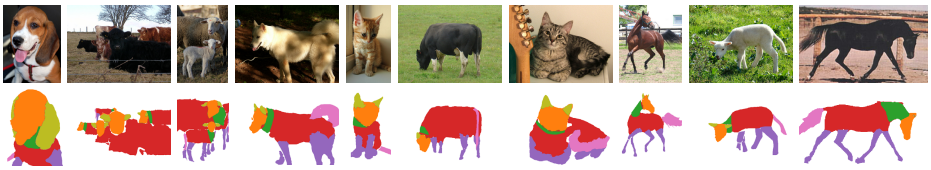


Figure 3: **Deep features visualization via PCA**: Applied on (a) ViTs and (b) CNN-ResNet models, each trained in a supervised manner for image classification, or in a self-supervised manner using DINO. We fed 18 images (examples in Fig. 10) to each model, extract feature maps from a given layer: activations for ResNet and keys, $\left[k_i^l\right]^{l\in\{2,5,8,11\}}$, for ViT. We visualize the first PCA components for each feature map, for the Dalmatian (Fig 10 left), the first component is shown on the top, while second-to-fourth components are shown as RGB images below.

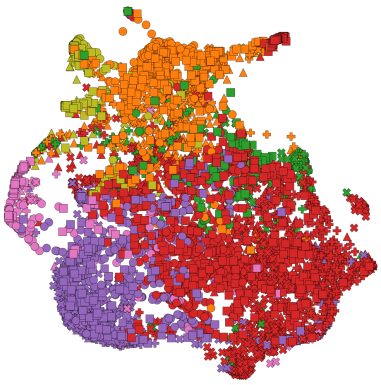
for image classification using ImageNet labels, and a *self-supervised ViT* (DINO-ViT), trained using a self-distillation approach [3]. We dive into the self-attention modules across layers, and empirically demonstrate that DINO-ViT features: (i) encode powerful high level information at high spatial resolution, i.e., capture semantic object parts at fine spatial granularity, and (ii) the encoded semantic information is shared across related object categories. We next provide intuitive visualizations of these properties, and empirically trace their origin to the *combination* of architecture and training supervision. In Sec. 5, we show these properties enable several applications, through which we quantitatively validate our observations.

Figure 3(a) shows a simple visualization of the learned representation by supervised ViT and DINO-ViT: for each model, we extract deep features (keys) from a set of layers, perform PCA, and visualize the resulting leading components. Fig. 3(b) shows the same visualization for two respective CNN-ResNet [19] models, one trained in a supervised manner, and the other using DINO. This simple visualization illustrates several fundamental differences between the resulting internal representations of each model.

Semantics vs. spatial granularity. One noticeable difference between CNN-ResNet and ViT is that CNNs trade spatial resolution with semantic information in the deeper layers, as shown in Fig. 3b: for the deepest layer, the feature maps are of very low resolution ($\times 32$ smaller), and thus provide poorly localized global semantic information. In contrast, ViT features' receptive field is the entire image from the very first layer – each token t_i^l attends to all other

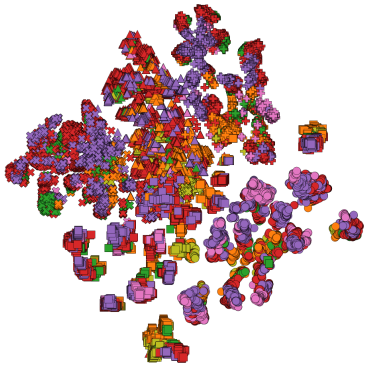


(a) Sample images and ground truth parts



(b) Self-Supervised ViT (DINO-ViT)

Torso
Neck
Head
Ears
Tail
Limbs
○ Cat
□ Dog
✣ Horse
⛛ Sheep
△ Cow



(c) Supervised ViT

Figure 4: **t-SNE visualization.** We take 50 images from PASCAL-Parts [7], 10 images from 5 animal categories; (a) representative images + ground-truth part segments. We extract ViT features (keys) from a self-supervised ViT (DINO-ViT), and a supervised ViT model. For each model, all keys are jointly projected to 2D using t-SNE. Each 2D point is colored according to its ground-truth *part*, while its shape represents the class. (b) DINO-ViT: keys are organized mainly by parts, across different object categories, while in (c) supervised ViT, they are grouped mostly by class, regardless of object parts.

tokens t_j^l. Also, ViT maintains the same spatial resolution through all layers. Thus, ViT features provide both fine-grained semantic information and high spatial resolution.

Furthermore, it is well known that the space of deep CNN-based features has a hierarchy of representation: early layers capture low-level elements such as edges or local texture patterns (shallow layers in Fig. 3b), while deeper layers gradually capture more high level concepts [37, 5, 43]. In contrast, we notice a different type of representation hierarchy in ViTs: shallow features are mostly dominated by the input position embeddings, while in deeper layers, this position bias is reduced in favor of more semantic features.

Semantic information across super-classes. Figure 3 exhibits the supervised ViT model (top) produces "noisier" features compared to DINO-ViT (bottom). To further contrast the two ViT representations, we employ t-SNE [38] to the keys of the last layer $\left[k_i^{11}\right]$, extracted from 50 animal images from PASCAL-Parts [7]. Figure 4 presents the 2D-projected keys. Intriguingly, the keys from a DINO-ViT show semantic similarity of body parts across different classes (grouped by *color*), while the keys from a supervised ViT display similarity within each class regardless of body part (grouped by *shape*). This demonstrates that while supervised ViT features emphasize *global* class information, DINO-ViT features have *local* semantic information akin to semantic object *parts*.

Different facets of ViT representation. So far we focused our discussion on the keys as 'ViT features'. Figure 5 shows slight differences in the representations of ViT facets; namely keys, queries, values and tokens. Whereas the keys are well separated in feature space (i.e., bear similarity to corresponding object *parts*), the other facets - queries, values and tokens, are less selective and bear similarity to less localized corresponding regions. We carefully ablate these observations in Sec. 5.3.

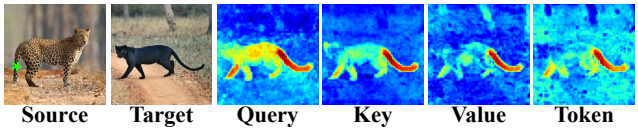


Figure 5: **Facets of ViT:** A DINO-ViT descriptor for the green point in the source image is compared via cosine similarity to all descriptors in the target image. We ablate different facets of ViT: keys, queries, value and tokens. The keys provide cleaner matches w.r.t other facets of ViT.

4. Deep ViT Features Applied to Vision Tasks

We demonstrate the effectiveness of Deep DINO-ViT Features as dense patch descriptors on a number of visual tasks: co-segmentation, part co-segmentation and semantic correspondences. We apply only simple, lightweight methodologies on the extracted features, without any additional training or fine-tuning. For full implementation details, see supplementary material (SM).

Co-segmentation. Our co-segmentation framework, applied to a set of N input images, comprises two steps, illustrated in Fig. 6(a-e):

1. *Clustering:* We treat the set of extracted features across all images and all spatial locations as a bag-of-features, and cluster them using an off-the-shelf clustering method. At this stage, the features are clustered into semantic common segments. As discussed in Sec. 3.1, the most prominent features' component distinguishes foreground and background, which ensures their separation.
2. *Voting:* We use voting to select clusters that both appear in most of the images and are salient. Saliency is measured per image, for each k_i^{11}, according to [CLS]

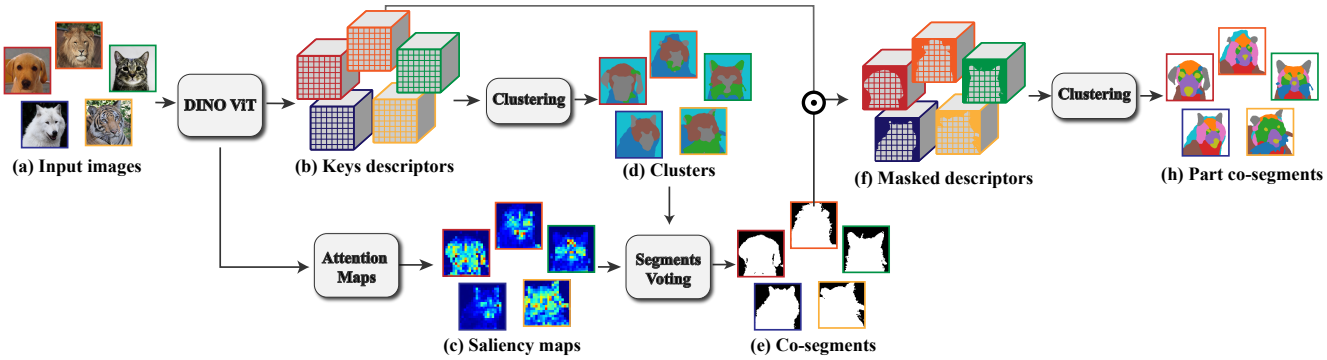


Figure 6: **Co-segmentation & part co-segmentation pipeline.** Input images (a) are fed separately to DINO-ViT to obtain (b) spatial dense features and (c) saliency maps (from ViT's self-attention maps). All the extracted features are clustered together. Each cluster is assigned as foreground or background via a saliency maps based voting process. Foreground segments form the co-segmentation result. The process is repeated on foreground features alone to yield the common parts (h).

attention [3]: $\mathsf{Sal}_i = \mathsf{SoftMax}\left(q^{11}_{[\mathsf{CLS}]} \cdot K^{11}\right)_i$

We further apply GrabCut [40] to refine the binary co-segmentation masks.

Part Co-segmentation. To further co-segment the foreground objects into common *parts*, we repeat the clustering and voting steps only on the foreground features, as illustrated in Fig. 6(f-h). By doing so, local descriptors of common semantic parts across images are clustered together, and the voting step filters clusters that are not common to most of the images. We further refine the segmentation masks using multi-label CRF [26].

In practice, we found k-means to perform well in our experiments, but other clustering methods can be easily plugged in. For co-segmentation, the number of clusters is automatically set using the elbow method [36], whereas for part co-segmentation, it is set to the desired number of object parts.

Our entire framework can be applied to a variety of object categories, and to arbitrary number of input images N, ranging from two to thousands of images. On small sets we apply random crop and flip augmentations for improved clustering stability.

Point Correspondences. We tackle the task of automatically finding corresponding points between two semantically related images. Semantic information is necessary, yet insufficient for this task. For example, matching points on the phone cords in Fig. 1(c-d), relying only on semantic information is ambiguous: all points on the cord are equally similar. We reduce this ambiguity in two manners:

1. *Positional Bias:* We want the descriptor to be position-aware. Earlier layers are biased towards positional representation (see Sec. 3.1); hence we use mid-layer features which provide a good trade-off between position and semantic information.
2. *Binning:* We incorporate context into each descriptor by integrating information from adjacent spatial features.

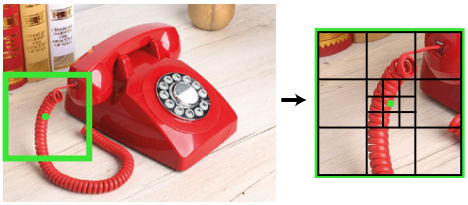


Figure 7: **Binning ViT features.** A descriptor associated with a green point in (a) aggregates its surrounding region (marked in green), according to the binning grid (b). The features within each bin are averaged and then concatenated to form our final patch descriptor.

This is done by applying log binning to each spatial feature, as illustrated in Fig. 7.

To automatically detect reliable matches between images, we adopt the notion of "Best Buddies Pairs" (BBPs) [12], i.e., we compute the cosine similarity between all descriptors pairs and keep only those who are mutual nearest neighbors. Formally, let $M = \{m_i\}$ and $Q = \{q_i\}$ be sets of binned descriptors from images I_M and I_Q respectively. The set of BBPs is given by:

$$\begin{aligned}\mathsf{BB}(M,Q) = \{(m,q) \mid m \in M,\ q \in Q,\\ \mathsf{NN}(m,Q) = q \wedge \mathsf{NN}(q,M) = p\}\end{aligned} \tag{1}$$

Where $\mathsf{NN}(m,Q)$ is the nearest neighbor of m in Q.

Resolution Increase. The spatial resolution of ViT features is inversely proportional to size of the *non-overlapping* patches, p_i. Our applications benefit from higher spatial feature resolution. We thus modified ViT to extract, at test time, *overlapping* patches, interpolating their positional encoding accordingly. Consequently, we get, without any additional training, ViT features at finer spatial resolution. Empirically, we found this method to work well in all our experiments. Further details appear in SM.

Data	Method	Training Set	$\mathcal{J}_m$	$\mathcal{P}_m$
MSRC [44]	Faktor et al. [15]	-	77.0	92.0
	Rubinstein et al. [41]	-	74.0	92.2
	SSNM [51]	COCO-SEG	81.9	95.2
	DOCS [29]	VOC2012	82.9	95.4
	CycleSegNet [28]	VOC2012	**87.2**	**97.9**
	Saliency Baseline	-	79.7	94.0
	Ours	-	**86.7**	**96.5**
Internet300 [41]	Rubinstein et al. [41]	-	57.3	85.4
	SSNM [51]	COCO-SEG	74.1	93.6
	DOCS [29]	VOC2012	72.5	93.5
	CycleSegNet [28]	VOC2012	80.4	-
	Li et al. [27]	COCO	**84.0**	**97.1**
	Saliency Baseline	-	59.1	85.0
	Ours	-	**79.5**	**94.6**
PASCAL-VOC [14]	Faktor et al. [15]	-	46.0	84.0
	SSNM [51]	COCO-SEG	71.0	94.9
	DOCS [29]	VOC2012	65.0	94.2
	CycleSegNet [28]	VOC2012	**75.4**	**95.8**
	Li et al. [27]	COCO	63.0	94.1
	Saliency Baseline	-	49.9	83.8
	Ours	-	**60.7**	**88.2**
PASCAL-CO	Faktor et al. [15]	-	41.4	79.9
	DOCS [29]	PASCAL	34.9	53.7
	SSNM [51]	COCO-SEG	74.2	94.5
	DINO ResNet	-	37.7	78.1
	Sup. ResNet	-	40.0	78.9
	Sup. ViT	-	39.9	69.7
	Saliency Baseline	-	75.0	93.1
	Ours	-	**79.5**	**94.7**

Table 1: **Co-segmentation evaluation:** We report mean Jaccard index $\mathcal{J}_m$ and precision $\mathcal{P}_m$ over all sets in each dataset, and compare to both supervised (training set specified), and unsupervised methods.

5. Results

5.1. Co-segmentation

We evaluate our performance on several *intra-class co-segmentation* datasets: (i) MSRC7 [44], has seven sets with ten images each. (ii) Internet300 [41], has three sets with a hundred images each. (iii) PASCAL-VOC [14] has twenty sets with dozens of images each.

Furthermore, to evaluate *inter-class co-segmentation*, we present a new dataset: PASCAL-Co-segmentation (PASCAL-CO), derived from PASCAL [14]. Our dataset has forty sets of six images, each from semantically related classes (e.g., car-bus-train, bird-plane). A sample set is shown in Fig. 8, the rest in SM.

We compare our *unsupervised* approach to: (i) state-of-the-art *supervised* methods, trained on large datasets with ground truth segmentation masks, SSNM [51], DOCS [29], Li et al. [27] and CycleSegNet [28]. (ii) *unsupervised* methods, Faktor et al. [15], and Rubinstein et al. [41]. Finally, we also compare to DINO-ViT saliency-based foreground segmentation method suggested in [3], and report it as 'Saliency Baseline'.

Figure 8: **PASCAL-CO for inter-class co-segmentation:** Each set contains images from semantically related classes.

Table 1 details results on all benchmarks: we report Jaccard Index ($\mathcal{J}_m$), which reflects both precision (covering the foreground) and accuracy (no foreground "leakage"), and in addition report mean precision ($\mathcal{P}_m$). Our method surpasses the unsupervised methods by a large margin, and is competitive to the supervised methods. In the *inter-class* scenario (PASCAL-CO), our method surpasses all other methods.

The simple 'Saliency Baseline' achieves impressive results, demonstrating the ability of the saliency map to capture foreground objects. However, our framework outperforms this baselines across all datasets, which quantifies the value of *jointly* processing all features across many images.

For PASCAL-CO, we compare ViT to ResNet, under each supervision (Tab. 1 bottom). Using DINO-ViT features surpass the rest by a margin, showing the superiority of this representation, which as shown in Sec. 3.1, is semantic, shared across super-classes, and of high granularity. More details are in SM.

5.2. Part Co-segmentation

We apply our part co-segmentation method to several datasets of different sizes: (i) Animal Faces HQ (AFHQ) [9] test set, containing images of different animal faces (ii) A subset of CelebA Human faces [31]. (iii) Three categories of CUB (birds) [49] test set. For (ii) and (iii), we use similar subsets to [20]. We further show results on in-the-wild image pairs taken from the Web.

Qualitative Results. Figures 1, 9, 10, 11 show sample results of our framework on the different datasets. In all cases, our results demonstrate the consistency of the common parts across images under significant variations in pose, appearance, and the number of objects in each image. The results provide further evidence of the semantic proximity of objects parts in ViT feature space across super-classes.

Quantitative Evaluation. Unsupervised part segmentation does not necessarily correspond to annotated object parts, hence we use a proxy evaluation task of semantic

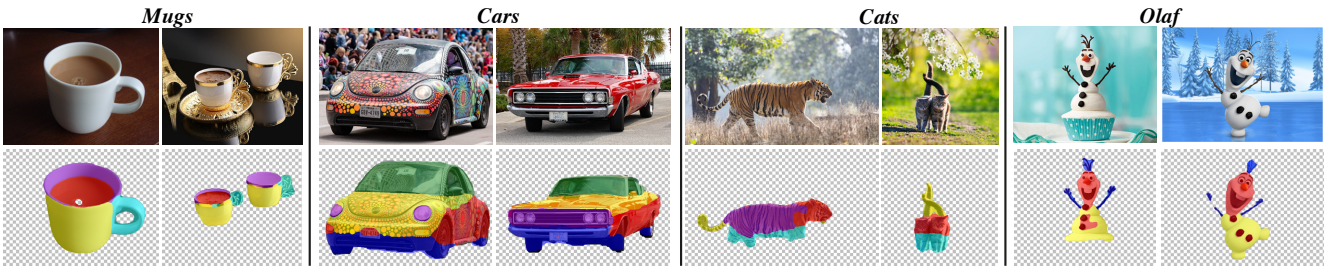


Figure 9: **Part Co-segmentation of Image *Pairs*:** Our method semantically co-segment common object parts given as little as two images as input. See SM for more examples.

Figure 10: **Part Co-segmentation on AFHQ:** Representatives from a 1.5K images set. More results included in SM.

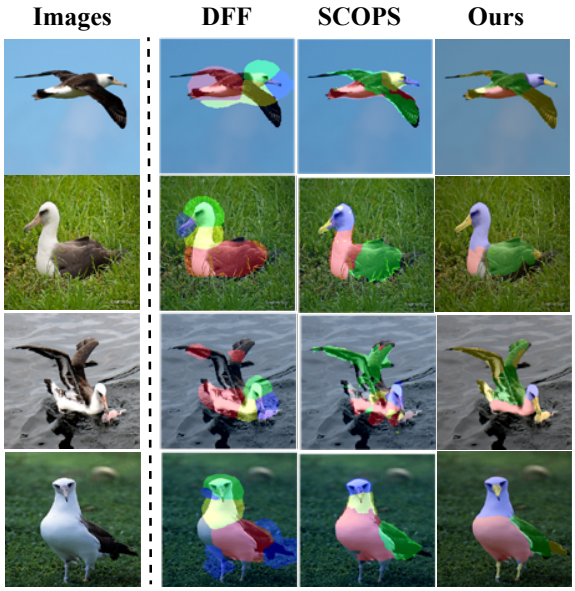


Figure 11: **Part co-segmentation comparison on CUB:** Comparing to recent methods [20, 10]. Our results are more semantically consistent across parts.

Method	CelebA $k=4$	CelebA $k=8$	CUB-1 $k=4$	CUB-2 $k=4$	CUB-3 $k=4$
ULD [47, 54]	-	40.82	30.12	29.36	28.19
DFF [10]	-	31.30	22.42	21.62	21.98
IMM [21]	19.42	**8.74**	N/A		
SCOPS [20] (w/o sal.)	46.62	22.11	18.50	18.82	21.07
SCOPS [20] (with sal.)	21.76	15.01	N/A		
Liu et al. [30]	15.39	12.26	18.15	17.54	**19.40**
Ours (DINO-ViT)	**11.36**	**10.74**	**17.14**	**14.67**	19.59
Ours (Sup. ViT)	12.83	12.74	N/A		

Table 2: **Landmark regression results:** We report mean error of landmark regression on CelebA [31] and CUB [49] data using the same protocol as [20] (lower is better). First three methods, listed for reference, are specially designed for landmarks discovery, IMM [21] specializes on faces.

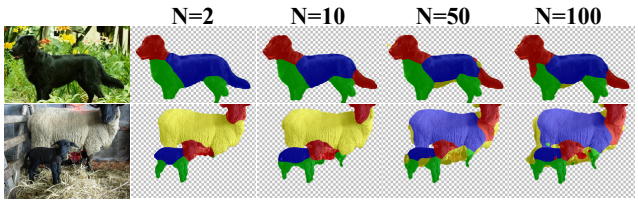


Figure 12: **Co-segmentation with varying number of images:** The two images (a), are part co-segmented with additional $N-2$ images from PASCAL Parts [7]. (b) Each column contains results for a different N. The part-segments' commonality improves when increasing N.

landmarks localization. We follow the protocol of [20] to quantitatively evaluate performance on CUB [49] and CelebA [31]: for each dataset, a linear regressor is trained to predict the ground truth landmarks given the centroids of the predicted part segments. We then report the error between the predicted and ground truth landmarks in Tab. 2. Our method achieves state-of-the-art performance on both datasets, compared to unsupervised part co-segmentation methods. Figure 11 shows our method produces more semantically coherent parts.

Additionally, we compare our performance to a supervised ViT on CelebA. Table 2 (bottom) clearly shows DINO-ViT's superiority over supervised-ViT. This result is expected, as supervised ViT mostly captures global class information, as discussed in Sec. 3.1.

Ablation. Fig. 12 illustrates the effect of the number of im-

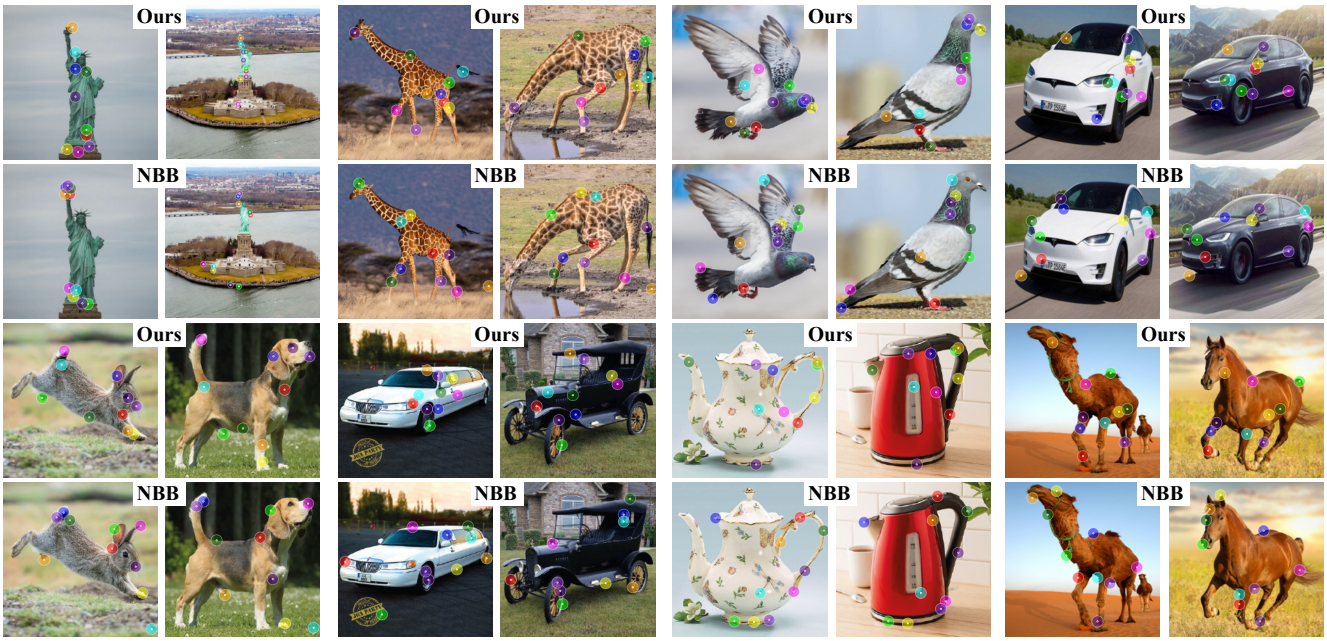


Figure 13: **Correspondences Comparison to NBB [1]:** On intra-class (top-row) and inter-class (bottom-row) scenarios. Our method is more robust to appearance, pose and scale variations. Full size results are available in SM.

Method	Backbone			PCK	
NBB [1]	VGG-19			26.98	
CATs [8]	ResNet-101			**61.43**	
Ours	DINO ViT	Layer	Facet	bins	w/o bins
		9	key	**56.48**	52.27
			query	54.96	49.35
			value	52.33	43.97
			token	56.03	50.14
		11	key	53.45	47.08
			query	52.35	42.64
			value	49.37	41.56
			token	50.34	46.09

Table 3: **Correspondence Evaluation on Spair71k:** We randomly sample 20 image pairs per category, and report the mean PCK across all categories ($\alpha = 0.1$); higher is better. We include a recent supervised method for reference.

ages on the resulting part co-segments. It appears using a small set of images with extreme variations can cause some segments that are semantically related to be clustered separately. This is remedied by (i) adding more images to the collection or (ii) augmenting the data (See Fig. 9).

5.3. Point Correspondences

Qualitative Results. We use binned DINO-ViT keys extracted from mid-level layer $l = 9$ to find semantic correspondences, as discussed in Sec. 4. Figure 13 shows comparison to NBB [1]. Our results are more robust to changes of appearance, pose and scale on intra- and inter-class pairs.

Quantitative Evaluation. We quantitatively evaluate on a subset of Spair71k [34], containing 360 randomly chosen pairs. The task is to find matching points in a target image to a given set of point in a source image. We calculate the Percentage of Correct Keypoint (PCK). A predicted keypoint is considered correct if it lies within a $\alpha \cdot \max(h, w)$ radius from the annotated keypoint, where (h, w) is the image size. We calculate the binned descriptors for the given keypoints in the source image, and find their nearest-neighbor in the target image. We compare to NBB[1] and a supervised baseline [8] for reference. Table 3 shows that our method outperforms NBB by a large margin, and closes the gap towards CATs.

Ablations. We used Spair71k to ablate our different design choices. We compared the different facets of DINO-ViT, namely, using queries, keys, values and tokens from layers 9 and 11, with and without binning. Table 3, empirically corroborates our observations on the sensitivity of the keys, and the representation hierarchy of ViT that trades positional-bias for semantic information in deeper layers (Sec. 3.1).

6. Conclusion

We provided new empirical observations on the internal features learned by ViT under different supervisions, and harnessed them for several real-world vision tasks. We demonstrated state-of-the-art results and new capabilities in representing objects at a fine spatial granularity across super-classes. In this work, we focused on simple, non-learnable methodologies, and we believe that our results hold great promise for considering deep ViT features as an alternative to deep CNN features. We plan to expand our research in this direction by adopting deep ViT features in more advanced, deep learning-based frameworks.

Acknowledgments: We would like to thank Miki Rubinstein and Meirav Galun for their insightful comments and discussion. This project received funding from ISF ***TODO: Tali?*** and the Carolito Stiftung. Dr Bagon is a Robin Chemers Neustein Artificial Intelligence Fellow.

References

[1] Kfir Aberman, Jing Liao, Mingyi Shi, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Neural best-buddies: Sparse cross-domain correspondence. *TOG*, 2018. 2, 8

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ECCV*, 2020. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 2, 3, 5, 6, 10, 11

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers: Github repository. https://github.com/facebookresearch/dino, 2021. 10

[5] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas. 2, 4

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2018. 2

[7] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. *CVPR*, 2014. 4, 7

[8] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Semantic correspondence with transformers. *NeurIPS*, 2021. 2, 3, 8

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *CVPR*, 2020. 6

[10] Edo Collins, Radhakrishna Achanta, and Sabine Süsstrunk. Deep feature factorization for concept discovery. *ECCV*, 2018. 2, 7

[11] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *ICLR*, 2019. 2

[12] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T. Freeman. Best-buddies similarity for robust template matching. *CVPR*, 2015. 5

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 2, 3

[14] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 6

[15] Alon Faktor and Michal Irani. Co-segmentation by composition. *ICCV*, 2013. 2, 6

[16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *CVPR*, 2016. 2

[17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*, 2019. 2

[18] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 3

[20] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. *CVPR*, 2019. 2, 6, 7

[21] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. *NeurIPS*, 2018. 7

[22] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: correspondence transformer for matching across images. *ICCV*, 2021. 3

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, 2016. 2

[24] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 10

[25] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus: Github repository. https://github.com/facebookresearch/faiss, 2017. 10

[26] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 2011. 5

[27] Bo Li, Zhengxing Sun, Qian Li, Yunjie Wu, and Anqi Hu. Group-wise deep object co-segmentation with co-attention recurrent neural network. *ICCV*, 2019. 2, 6

[28] Guankai Li, Chi Zhang, and Guosheng Lin. Cyclesegnet: Object co-segmentation with cycle refinement and region correspondence. *TIP*, 2021. 2, 6

[29] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. *ACCV*, 2018. 2, 6

[30] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. *CVPR*, 2021. 2, 7

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015. 6, 7, 10

[32] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *NeurIPS*, 2016. 2

[33] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. *ECCV*, 2018. 2

[34] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *CoRR*, 2019. 8

[35] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan

Yang. Intriguing properties of vision transformers. *NeurIPS*, 2021. 1, 2

[36] Andrew Ng. Clustering with the k-means algorithm. *Machine Learning*, 2012. 5

[37] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 2, 4

[38] Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. opentsne: a modular python library for t-sne dimensionality reduction and embedding. *bioRxiv*, 2019. 4

[39] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *NeurIPS*, 2021. 2

[40] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *TOG*, 2004. 5

[41] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. *CVPR*, 2013. 2, 6, 10

[42] Jose C. Rubio, Joan Serrat, Antonio López, and Nikos Paragios. Unsupervised co-segmentation through region matching. *CVPR*, 2012. 2

[43] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T. Freeman, and Tali Dekel. Semantic pyramid for image generation. *CVPR*, 2020. 2, 4

[44] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 2006. 6

[45] 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. *BMVC*, 2021. 2

[46] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 3

[47] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. *ICCV*, 2017. 7

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CVPR*, 2018. 2

[49] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 6, 7

[50] Ross Wightman. Pytorch image models. *GitHub repository*, 2019. 10, 11

[51] Kaihua Zhang, Jin Chen, Bo Liu, and Qingshan Liu. Deep object co-segmentation via spatial-semantic network modulation. *AAAI*, 2020. 2, 6

[52] Richard Zhang. Making convolutional networks shift-invariant again. *ICML*, 2019. 2

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 2

[54] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. *CVPR*, 2018. 7

A. Implementation Details

In all our applications (unless specified otherwise) we use `dino_vits8` model from the official DINO Github repository [3, 4], with `stride=4` (see Sec. B).

Co-segmentation parameters (§ 5.1). We extracted the keys from the last layer (11^{th} starting from 0), concatenated all the heads to receive a descriptor for each patch. We used the FAISS library [24, 25] for computing k-means. In the co-segmentation experiments our elbow coefficient is 0.975, saliency threshold is 0.065, majority percentage is 75%. We resize the input images to have the shorter edge of size 320[pix].

Global Outlier Filtering One of the challenges in the Internet300 [41] dataset is handling images that do not contain the common object at all. We term these *global outlier images*, and filter them automatically before applying the co-segmentation pipeline using the descriptor of the [CLS] token. We compute the average of all the [CLS] descriptors on the entire set of images, and reject images that have cosine similarity lower than 0.7 from the average descriptor.

Co-segmentation and Part Co-segmentation Ablations. Supervised ViT weights are from `timm` repository [50]. We used keys from the 9^{th} layer because they exhibited better part separation than the 11^{th} layer, giving supervised ViT a fair chance. We used `vit_small_patch16_224` with `stride=4`. In saliency baseline we used a saliency threshold of 0.04. DINO and supervised `ResNet-50` weights are from DINO and `timm` repositories respectively. In PASCAL-Co ablations for `ResNet-50` we replace the last three strides with dilation to receive high resolution feature maps, as if features were computed at `stride=4` of the input resolution. All models are trained on ImageNet data.

Part Co-segmentation parameters (§5.2). We use the same parameters as co-segmentation application. For CelebA [31], we choose the salient segments based if there average distance from the center of the image was under 0.2, and if their compactness was higher than 0.5.

Part Co-Segmentation of Image Pairs. (Fig. 9) We present our part co-segmentation results in an extreme setting – operating on two images under significant variations of quantity, background clutter, pose, scale and appearance. We use flip and random-crop (95% of the original images) augmentations to compensate for the low number of images. We also introduce three clustering stages instead of two – one for fg/bg separation, one for removing uncommon foreground objects and one for part segmentation. This extreme setting is sensitive to hyper-parameters, but we found using 40 random-crop augmentations, and elbow coefficient of 0.94 works well for most cases.

Correspondence parameters (§5.3). For compatibility with NBB we resized the images to 224×224. We use saliency threshold of 0.05. We use log-binning with 2 hierarchies (17 bins, like shown in Fig. 7).

t-SNE. (Fig. 4) We used the same configurations as mentioned previously, besides these modifications: we used

Architecture	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
ViT-S/16	61.8	60.2	63.4
ViT-B/16	62.3	60.7	63.9
ViT-S/8	69.9	66.6	73.1
ViT-B/8	71.4	67.9	74.9
Ours	**72.2**	**67.9**	**76.5**

Table 4: DAVIS 2017 Video Object Segmentation.

Layer 11 in supervised ViT and `stride=8` in both models.

PCA. (Fig. 3) We used `dino_vits16` and `vit_small_patch16_224` models with `stride=8`. We resized the input images to size 224×224.

B. Resolution Increase (§4)

We use `timm` repository [50] for ViT architecture and supervised weights, and [3] for DINO-ViT weights. We increase the resolution of ViT features maps by altering the phase of patch preparation. Instead of taking non-overlapping patches we take *overlapping patches*. In practice, the separation to patches and linear embedding is done by passing the image through a single convolution layer, with stride that equals the patch size and number of out channels as the embedding dimension. We alter the *stride* of this convolution layer to achieve overlapping patches. For example, using `stride=8` for a ViT trained with patch size 16 will increase the ViT feature's resolution times two. We assume the input size $\{H_{in}, W_{in}\}$ is divided by the patch size without remainder. If that is not the case, we remove the remainder pixels from the image. The output size is given by:

$$H_{\text{out}} = \frac{H_{\text{in}} - \text{patch\_size}}{\text{stride}} + 1$$
$$W_{\text{out}} = \frac{W_{\text{in}} - \text{patch\_size}}{\text{stride}} + 1$$

DAVIS Label Propagation. We empirically show the usefulness of test-time resolution increase by applying it to one of the applications shown in [3] - using pre-trained DINO features for DAVIS label propagation. We used a `dino_vits8` model with `stride=4`. Table 4 exhibits a significant improvement in results when using our alteration, and exhibit results even better than `dino_vitb8`.

Spair71k Keypoint Matching. In Tab. 5 we ablate our keypoint matching method with and without resolution increase. Evidently, increasing resolution enables higher spatial granularity which improves the performance of the method.

category	**NBB**	**CATs**	**Stride 4**	**Stride 8**
aeroplane	0.44	0.57	**0.69**	0.64
bicycle	0.28	0.48	**0.50**	0.49
bird	0.67	**0.89**	0.82	0.78
boat	0.12	0.39	**0.47**	0.43
bottle	0.17	**0.44**	0.37	0.33
bus	0.20	**0.63**	0.42	0.36
car	0.28	**0.60**	0.53	0.52
cat	0.30	0.65	**0.66**	0.62
chair	0.20	0.34	**0.45**	0.39
cow	0.29	0.73	**0.75**	0.63
dog	0.37	**0.65**	**0.65**	0.63
horse	0.13	**0.60**	0.46	0.38
motorbike	0.51	**0.80**	0.69	0.68
person	0.14	**0.66**	0.48	0.38
pottedplant	0.15	**0.48**	0.44	0.44
sheep	0.11	**0.70**	0.65	0.62
train	0.23	**0.83**	0.54	0.45
tvmonitor	0.26	**0.62**	0.59	0.55
all	0.27	**0.61**	0.56	0.52

Table 5: Spair71k keypoint matching with different strides