Deep ViT Features as Dense Visual Descriptors

Shir Amir[1], Yossi Gandelsman[2], Shai Bagon[1], and Tali Dekel[1]

[1] Dept. of Computer Science and Applied Math, The Weizmann Inst. of Science
[2] Berkeley Artificial Intelligence Research (BAIR)

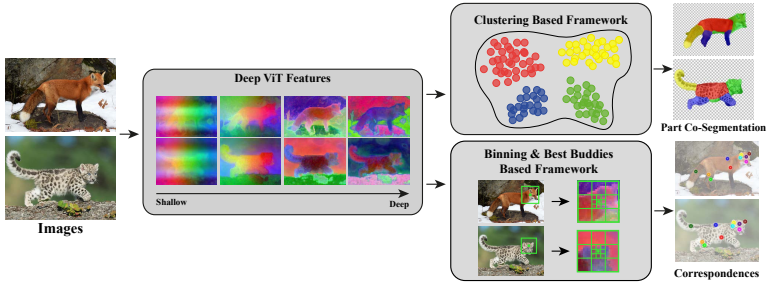

Fig. 1. Based on our new observations on deep ViT features, we devise *lightweight zero-shot* methods to solve fundamental vision tasks (e.g. part co-segmentation and semantic correspondences). Our methods are applicable even in challenging settings where the images belong to different classes (e.g. fox and leopard).

Abstract. We study the use of deep features extracted from a pre-trained Vision Transformer (ViT) as dense visual descriptors. We observe and empirically demonstrate that such features, when extracted from a self-supervised ViT model (DINO-ViT), exhibit several striking properties, including: (i) the features encode powerful, well-localized semantic information, at high spatial granularity, such as object *parts*; (ii) the encoded semantic information is *shared across related, yet different object categories*, and (iii) positional bias changes gradually *throughout the layers*. These properties allow us to design simple methods for a variety of applications, including co-segmentation, part co-segmentation and semantic correspondences. To distill the power of ViT features from convoluted design choices, we restrict ourselves to *lightweight zero-shot* methodologies (e.g., binning and clustering) applied directly to the features. Since our methods require no additional training nor data, they are readily applicable across a variety of domains. We show by extensive qualitative and quantitative evaluation that our simple methodologies achieve competitive results with recent state-of-the-art *supervised* methods, and outperform previous unsupervised methods by a large margin. Code is available in `dino-vit-features.github.io`.

Keywords: ViT, deep features, zero-shot methods

1 Introduction

"Deep Features" – features extracted from the activations 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 [49]) 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 numerous visual tasks, while demonstrating better robustness to occlusions, adversarial attacks and texture bias compared to CNN-based models [37]. This raises the following questions: Do these properties reflect on the internal representations 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]. In contrast to existing methods, which mostly focus on the features from the deepest layer [3, 48, 54], we dive into the self-attention modules, and consider the various facets (tokens, queries, keys, values) *across different layers*. We observe and empirically demonstrate that DINO-ViT features: (i) encode powerful high-level information at high spatial resolution, i.e., capture semantic object *parts*, (ii) this encoded semantic information is *shared across related, yet different object classes*, and (iii) positional information gradually decreases *throughout* layers, thus the intermediate layers encode position information as well as semantics. We demonstrate that these properties are not only due to the ViT architecture but also significantly influenced by the training supervision.

Relying on these observations, we unlock the effectiveness of DINO-ViT features by considering their use in a number of fundamental vision tasks: co-segmentation, part co-segmentation, and semantic point correspondences. Moreover, equipped with our new observations, we tackle the task of part co-segmentation in a *challenging unconstrained setting* where neither the number of input images, nor their domains are restricted. We further present how our part co-segmentation can be applied to videos. To the best of our knowledge, we are the first to show results of part co-segmentation in such challenging cases (Fig. 6). We apply *simple, zero-shot* methodologies to deep ViT features for all these tasks, which do not require further training. Deliberately avoiding large-scale learning-based models showcases the effectiveness of the learned DINO-ViT representations. We demonstrate that without bells and whistles, DINO-ViT features are already powerful enough to achieve competitive results compared to

state-of-the-art models specifically designed and trained for each individual task. We thoroughly evaluate our performance qualitatively and quantitatively.

To conclude, our key contributions are: (i) We uncover surprising *localized semantic information*, far beyond saliency, readily available in ViT features. (ii) Our new observations give rise to *lightweighted zero-shot* methodologies for tackling co- and part co-segmentation as well as semantic correspondences. (iii) We are the first to show part co-segmentation in *extreme settings*, showing how objects can be consistently segmented into parts across different categories, and across a variety of image domains, for some of which training data is scarce.

2 Related Work

CNN-based Deep Features. Features of pre-trained CNNs are a cornerstone for various vision tasks from object detection and segmentation [19, 5], to image generation [46, 17]. These representations were shown to align well with human perception [17, 26, 58, 35] 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) [40, 4]. Nevertheless, they exhibit a strong bias towards texture [18], and lack positional information due to their shift equivariance [57]. Moreover, their restricted receptive field [34] makes them capture mostly local information and ignore long-range dependencies [53]. Here, we study the deep features of a less restrictive architecture - the Vision Transformer, as an alternative.

Vision Transformer (ViT). Vision Transformers [13] have recently been used as powerful CNN alternatives. ViT-based models achieve impressive results in a variety of visual tasks [13, 7, 2], while demonstrating better robustness to occlusions, adversarial attacks, and texture bias compared to CNN-based models [37].

In particular, Caron et al. [3] presented DINO-ViT – a ViT model trained without labels, using a self-distillation approach. They observed that the attention heads of this model attend to salient foreground regions in an image. They further showed the effectiveness of DINO-ViT features for several tasks that benefit from this property, including image retrieval and object segmentation.

Recent works follow this observation and utilize these features for object discovery [48, 54], semantic segmentation [20] and category discovery [52]. All these works treat pre-trained DINO-ViT as a black-box, only considering features extracted from it's last layer, and their use as global or figure/ground-aware representations. In contrast, we examine the continuum of Deep ViT features *across layers*, and dive into the different representations inside each layer (e.g. the keys, values, queries of the attention layers). We observe *new* properties of these features besides being aware to foreground objects, and put these observations to use by solving fundamental vision tasks.

Concurrently, [42, 11, 37] study theoretical aspects of the underlying machinery, aiming to analyze how ViTs process visual data compared to CNN models. Our work aims to bridge the gap between better understanding Deep ViT representations and their use in real-world vision tasks in a zero-shot manner.

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, 44, 45] for this task. Later, CNN-based methods applied supervised training [31] or fine-tuning [56, 29, 30] 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 new *inter-class* scenarios. We, however, show a *lightweight 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* among the images. Recent methods [24, 32, 9] 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. In contrast, we utilize a pre-trained self-supervised ViT to solve this task, and achieve competitive performance to the methods above. Due to the zero-shot nature of our approach, we are able to apply part co-segmentation *across classes*, and on domains that lack training supervision (see Fig. 6). To the best of our knowledge, we are the first to address such challenging scenarios.

Semantic Correspondences. Given a pair of images, the task is to find semantically corresponding points between them. Aberman et al. [1] propose a sparse correspondence method for inter-class scenarios leveraging pre-trained CNN features. Recent *supervised* methods employ transformers for dense correspondence in images from the same scene [50, 25]. Cho et al. [7] use transformers for semantic point correspondences by training directly on annotated point correspondences. We show that utilizing ViT features in a *zero-shot* manner can be competitive to *supervised* methods while being more robust to different pose and scale than previous unsupervised methods.

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):

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

where $T^l = \left[t_0^l, \ldots, t_n^l\right]$ are the output tokens for layer l.
In each MSA block, tokens are linearly projected into queries, keys and values:

$$q_i^l = W_q^l \cdot t_i^{l-1}, \quad k_i^l = W_k^l \cdot t_i^{l-1}, \quad v_i^l = W_v^l \cdot t_i^{l-1} \tag{2}$$

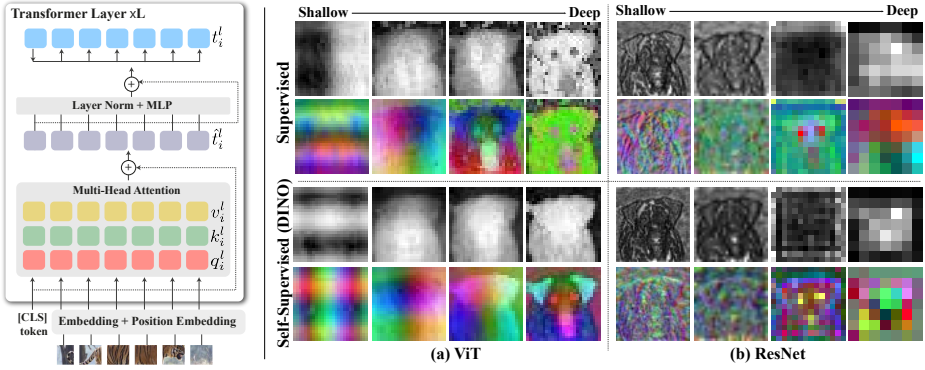

Fig. 2. *ViT Architecture (Left).* An image is split into n non-overlapping patches and gets a `[CLS]` token. These patches are embedded, added positional embeddings and passed through transformer layers. Each patch is directly associated with a set of features in each layer: a key, query, value and token; each can be used as patch descriptors. *Deep features visualization via PCA (Right):* Applied on supervised and self-supervised (a) ViTs and (b) CNN-ResNet models. We fed 18 images from AFHQ [8] to each model, extract features from a given layer, and perform PCA on them. For each model, we visualize PCA components at each layer, for an example image (Dalmatian dog in Fig. 7 left): the first component is shown on the top, while second-to-fourth components are shown as RGB images below. ResNet PCA is upsampled for visualization purposes.

which are then fused using multihead self-attention. Figure 2 Left illustrates this process, for full details see [13]. Besides the initial image patches sampling, ViTs have no additional spatial sampling; hence, each image patch p_i is *directly* associated with a set of features: $\{q_i^l, k_i^l, v_i^l, t_i^l\}$, including its query, key, value and token, at each layer l, respectively. We next focus our analysis on using the *keys* as 'ViT features'. We justify this choice via ablation in Sections 5.2 & 5.3.

3.1 Properties of ViT's Features

We focus on two pre-trained ViT models, both have the same architecture and training data, but differ in their training supervision: a *supervised ViT*, trained for image classification using ImageNet labels [13], and a *self-supervised ViT* (DINO-ViT), trained using a self-distillation approach [3]. We next provide qualitative analysis of the internal representations learned by both models, and empirically originate their properties 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 2 Right (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. Figure 2 Right (b) shows the same visualization for two respective CNN-ResNet [21] models trained using the same two supervisions as the

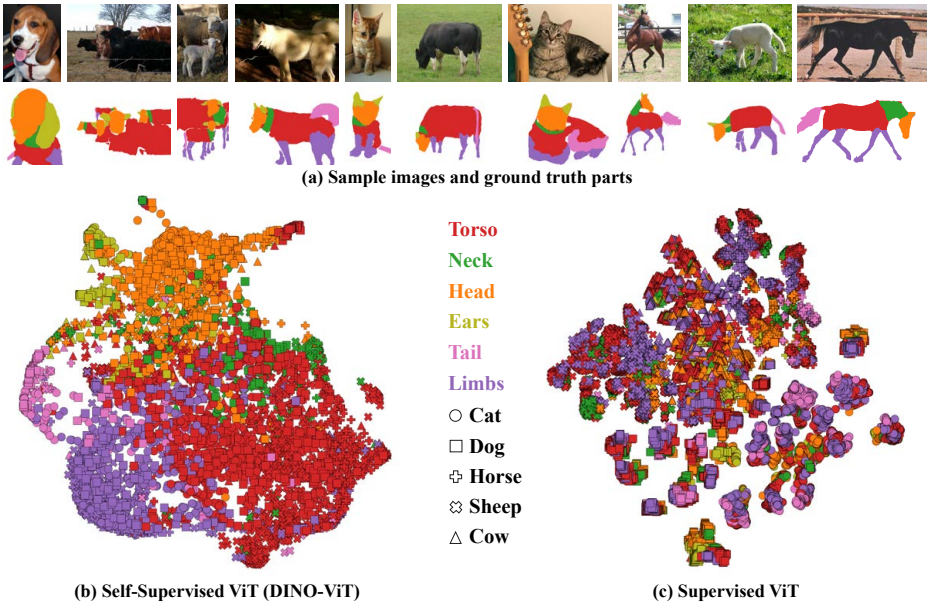

Fig. 3. *t-SNE visualization.* We take 10 images from 5 animal categories from PASCAL-Parts [6]. (a) shows representative images and ground-truth part segments. For each image we extract ViT features from DINO-ViT and a supervised ViT. For each model, all features are jointly projected to 2D using t-SNE [41]. Each 2D point is colored according to its ground-truth *part*, while its shape represents the class. In (b) DINO-ViT features are organized mainly by parts, across different object categories, while in (c) supervised ViT features are grouped mostly by class, regardless of object parts.

ViT models: image classification, and DINO [3]. This simple visualization shows fundamental differences between the 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. 2 Right (b): the feature maps in the deepest layer have very low resolution ($\times 32$ smaller than the input image), and thus provide poorly localized semantic information. In contrast, ViT maintains the same spatial resolution through all layers. Also, the receptive field of ViT is the entire image in all layers – each token t_i^l attends to all other tokens t_j^l. Thus, ViT features provide fine-grained semantic information *and* higher spatial resolution.

Representations across layers. 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 textures (shallow layers in Fig. 2 Right (b)), while deeper layers gradually capture more high level concepts [40, 4, 46]. In contrast, we notice a different type of representation hierarchy in ViTs: *Shallow features mostly contain positional information*, while in deeper layers, this is reduced in

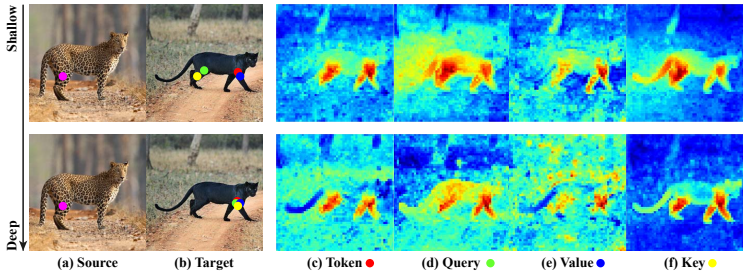

Fig. 4. *Facets of ViT:* We compute the similarity between a feature associated with the magenta point in the source image (a) to all features in the target image (b). We do this for intermediate features (top row) and features from the last layer (bottom row). (c-f) are the resulting similarity maps, when using different facets of DINO-ViT as features: tokens, queries, values and keys. Red indicates higher similarity. For each facet, the closest point in the target image is marked with a unique color, specified near the facet name. The keys (f) have cleaner similarity map compared to other facets.

favor of more semantic features. For example, in Fig. 2 Right (a) the deep features distinguish dog features from background features, while the shallow features are gathered mostly based on their spatial location. Interestingly, intermediate ViT features contain both positional and semantic information.

Semantic information across super-classes. Figure 2 Right (b) exhibits the supervised ViT model (top) produces "noisier" features compared to DINO-ViT (bottom). To further contrast the two ViTs, we employ t-SNE [41] to the keys of the last layer $\left[k_i^{11}\right]$, extracted from 50 animal images from PASCAL-Parts [6]. Figure 3 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 spatial features emphasize *global* class information, DINO-ViT features have *local* semantic information resembling semantic object *parts*.

Different facets of ViT representation. So far, we focused on using keys as 'ViT features'. However, ViT provides different facets that are also directly associated with each image patch (Fig. 2 Left). We empirically observe slight differences in the representations of ViT facets, as shown in Fig. 4. In particular, we found the keys to provide a slightly better representation, e.g., they depict less sensitivity to background clutter than the other facets. In addition, both keys and queries posses more positional bias in intermediate layers than values and tokens.

4 Deep ViT Features Applied to Vision Tasks

We demonstrate the effectiveness of deep DINO-ViT features as local patch descriptors several visual tasks. We *deliberately* apply only simple, lightweight

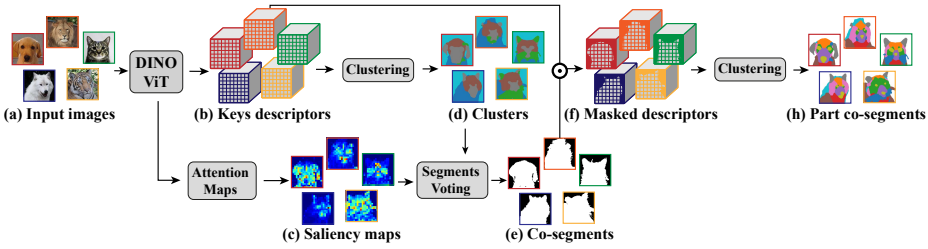

Fig. 5. *Co-segmentation & part co-segmentation pipeline.* Input images (a) are fed separately to DINO-ViT to obtain (b) spatial dense descriptors and (c) saliency maps (from the ViT's self-attention maps). All the extracted descriptors are clustered together (d). Each cluster is assigned as foreground or background via a saliency maps based voting process. Foreground segments form the co-segmentation results (e). The process is repeated on foreground features (f) alone to yield the common parts (h).

methodologies on the extracted features, without any additional training nor fine-tuning, to showcase the effectiveness of DINO-ViT representations. For full implementation details, see supplementary material (SM).

Co-segmentation. Our co-segmentation approach, applied to a set of N input images, comprises of two steps, followed by GrabCut [43] to refine the binary co-segmentation masks, as illustrated in Fig. 5(a-e):

1. *Clustering:* We treat the set of extracted descriptors across all images and all spatial locations as a bag-of-descriptors, and cluster them using k-means. At this stage, the descriptors are clustered into semantic common segments. As illustrated in Fig. 2 Right, the most prominent features' component distinguishes foreground and background, which ensures their separation. The result of this stage is K clusters that induce segments in all images.
2. *Voting:* We use a simple voting procedure to select clusters that are salient and common to most of the images. Let $\mathsf{Attn}_i^{\mathcal{I}}$ be the mean `[CLS]` attention of selected heads in the last layer in image $\mathcal{I}$ of patch i. Let $S_k^{\mathcal{I}}$ be the set of all patches in image $\mathcal{I}$ belonging to cluster k. The saliency of segment $S_k^{\mathcal{I}}$ is:

$$\mathsf{Sal}\left(S_k^{\mathcal{I}}\right) = \frac{1}{|S_k^{\mathcal{I}}|} \sum_{i \in S_k^{\mathcal{I}}} \mathsf{Attn}_i^{\mathcal{I}} \tag{3}$$

Each segment votes for the saliency of the cluster k:

$$\mathsf{Votes}\,(k) = \mathbb{1}_{\left[\sum_{\mathcal{I}} \mathtt{Sal}\left(S_k^{\mathcal{I}}\right) \geq \tau\right]} \tag{4}$$

For some threshold τ. A cluster k is considered "foreground" iff its $\mathsf{Votes}\,(k)$ is above percentage p of all the images.

Part Co-segmentation. To further co-segment the foreground objects into common *parts*, we repeat the clustering step only on foreground descriptors, see

Fig. 5(f-h). By doing so, descriptors of common semantic parts across images are grouped together. We further refine the part masks using multi-label CRF [27]. In practice, we found k-means to perform well, but other clustering methods (e.g. [39, 16]) can be easily plugged in. For co-segmentation, the number of clusters is automatically set using the elbow method [38], whereas for part co-segmentation, it is set to the desired number of object parts. Our method 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 (see SM for more details).

Point Correspondences. Semantic information is necessary yet insufficient for this task. For example, matching points on an animal's tail in Fig. 1, relying only on semantic information is ambiguous: all points on the tail are equally similar. We reduce this ambiguity in two manners:

1. *Positional Bias:* We want the descriptors to be position-aware. Features from earlier layers are more sensitive to their position in the image (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. This is done by applying log-binning to each spatial feature, as illustrated in Fig. 1.

To automatically detect reliable matches between images, we adopt the notion of "Best Buddies Pairs" (BBPs) [12], i.e., we only keep descriptor pairs which 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 thus:

$$\mathsf{BB}(M, Q) = \{(m, q) \mid m \in M,\ q \in Q,\ \mathsf{NN}(m, Q) = q \wedge \mathsf{NN}(q, M) = m\} \tag{5}$$

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

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 modify 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.

5 Results

5.1 Part Co-segmentation

Challenging small sets. In Fig. 6, we present several image pairs collected from the web. These examples pose challenge due to different appearance (e.g. cars, phones), different classes (e.g. bicycle-unicycle, cat-tiger) and belonging to domains that are difficult to accommodate training sets for (e.g. pikachu, olaf). Our

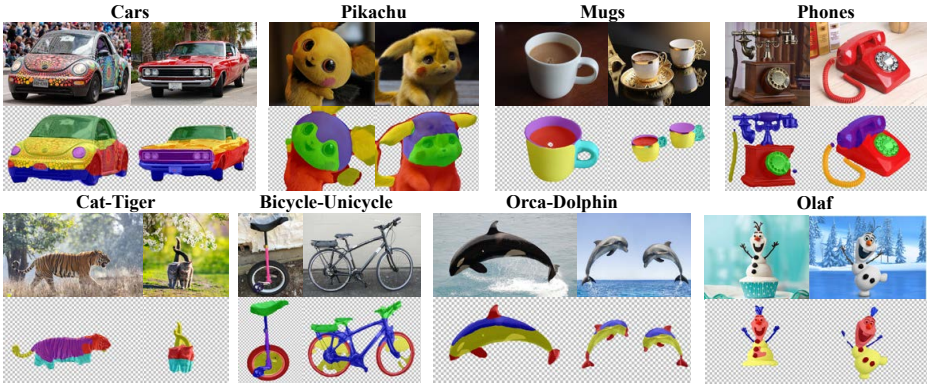

Fig. 6. *Part Co-segmentation of Image* Pairs*:* Our method semantically co-segments common object parts given as little as two input images. See the SM for more examples.

Fig. 7. *Part Co-segmentation on AFHQ:* We apply our method on the test set of AFHQ [8] containing 1.5K images of different animal faces. More results are in SM.

zero-shot method manages to provide semantically consistent part segments for each image pair. For example, in the bicycle-unicycle example the tires, spokes, chassis and saddle parts are consistently found. To the best of our knowledge, we are the first to handle such challenging cases.

Video part Co-segmentation. We extend our framework to work on videos by applying it to frames of a single video. Since DINO-ViT features are consistent across video frames, applying our observations to video co-segmentation yields temporally consistent parts. To the best of our knowledge, we are the first to apply part co-segmentation on videos. We include multiple examples in the SM.

Inter-class results. In Fig. 7 we apply our part co-segmentation with $k = 10$ parts on AFHQ [8] test set, containing 1.5K images of different animal faces. Our method provides consistent parts across *different* animal classes, e.g. ears marked in orange, forehead marked in blue, whiskers marked in purple, etc.

CUB [55] evaluation. Following [24, 9], we evaluate performance on CUB [55] test set, which contains 5K images of different bird species. Following [24], we measure the key-point regression error between the predicted and ground truth

Fig. 8. *Part co-segmentation comparison on CUB:* We show results on randomly chosen images from CUB [55]. Our results are more semantically consistent across parts than the *supervised* SCOPS [24] and are competitive to the *supervised* Choudhury et al. [9].

Method	key-point regression ↓ CUB-01	CUB-02	CUB-03	FG-NMI ↑	FG-ARI ↑	NMI ↑	ARI ↑
supervised							
SCOPS [24]‡ (model)	18.3	17.7	17.0	39.1	17.9	24.4	7.1
Huang and Li [23]†	15.1	17.1	15.7	-	-	26.1	13.2
Choudhury et al.[9]‡	**11.3**	15.0	**10.6**	**46.0**	**21.0**	**43.5**	**19.6**
unsupervised							
ULD [51, 59]	30.1	29.4	28.2	-	-	-	-
DFF [10]	22.4	21.6	22.0	32.4	14.3	25.9	12.4
SCOPS [24] (paper)	18.5	18.8	21.1	-	-	-	-
Ours	17.1	**14.7**	19.6	39.4	19.2	38.9	16.1

Table 1. *Part Co-segmentation results:* We report mean error of landmark regression on three CUB [55] test sets, and NMI and ARI [9] measures on the entire CUB test set. All methods predict $k = 4$ parts. † method uses image-level supervision, ‡ methods use ground truth foreground masks as supervision.

landmarks in Tab. 1 on three test sets from CUB [55]. In addition, we follow [9] treating the part segments as clusters, and report NMI and ARI. FG-NMI and FG-ARI disregard the background part as a cluster. Our method surpasses unsupervised methods *by a large margin*, and is competitive to [9] which is *supervised* by foreground masks. Figure 8 shows our method produces more semantically coherent parts, with similar quality to [9]. Further evaluation on the CelebA [33] dataset is available in the SM.

5.2 Co-segmentation

We evaluate our performance on several *intra-class* co-segmentation datasets of varying sizes - MSRC7 [47], Internet300 [44] and PASCAL-VOC [14].Furthermore, to evaluate *inter-class* co-segmentation, we compose a new dataset from PASCAL [14] images, named "PASCAL Co-segmentation" (PASCAL-CO). Our

Method	Training Set	MSRC [47]		Internet300 [44]		PASCAL -VOC [14]		PASCAL -CO	
		$\mathcal{J}_m$	$\mathcal{P}_m$	$\mathcal{J}_m$	$\mathcal{P}_m$	$\mathcal{J}_m$	$\mathcal{P}_m$	$\mathcal{J}_m$	$\mathcal{P}_m$
supervised									
SSNM [56]	COCO-SEG	81.9	95.2	74.1	93.6	71.0	94.9	74.2	94.5
DOCS [31]	VOC2012	82.9	95.4	72.5	93.5	65.0	94.2	34.9	53.7
CycleSegNet [30]	VOC2012	**87.2**	**97.9**	80.4	-	**75.4**	**95.8**	-	-
Li et al. [29]	COCO	-	-	**84.0**	**97.1**	63.0	94.1	-	-
unsupervised									
Hsu et al.[22]	-	-	-	69.8	92.3	60.0	91.0	-	-
DeepCO3 [28]	-	54.7	87.2	53.4	88.0	46.3	88.5	37.3	74.1
TokenCut[54]	-	81.2	94.9	65.2	91.3	57.8	90.6	75.8	93.0
Faktor et al.[15]	-	77.0	92.0	-	-	46.0	84.0	41.4	79.9
Rubinstein et al.[44]	-	74.0	92.2	57.3	85.4	-	-	-	-
Ours	-	86.7	96.5	79.5	94.6	60.7	88.2	**79.5**	**94.7**

Table 2. *Co-segmentation evaluation:* We report mean Jaccard index $\mathcal{J}_m$ and precision $\mathcal{P}_m$ over all sets in each dataset. We compare to unsupervised methods [15, 44] and methods supervised with ground truth segmentation masks [56, 31, 30, 29].

	DINO Saliency Baselines		Sup. Saliency Baselines		Ours			
	ViT	ResNet	ViT	ResNet	Keys	Tokens	Queries	Values
$\mathcal{J}_m$	75.0	37.7	39.9	40.0	**79.5**	69.2	72.7	49.2
$\mathcal{P}_m$	93.1	78.1	69.7	78.9	**94.7**	90.68	91.7	83.3

Table 3. *Co-segmentation ablation:* on PASCAL-Co for saliency baselines and our method using different ViT facets. Our method surpasses all baselines, and our choice of keys yields better performance than default chosen DINO-ViT tokens.

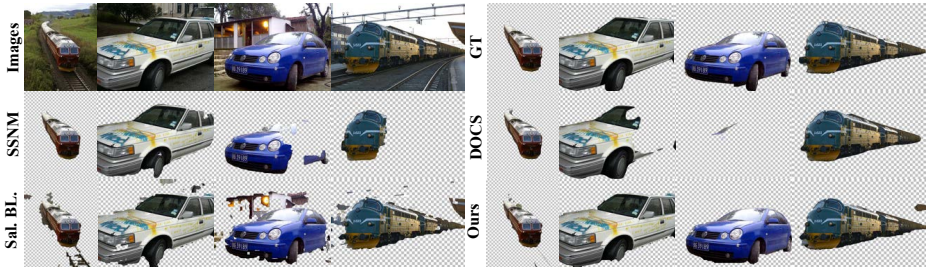

Fig. 9. *PASCAL-CO for inter-class co-segmentation:* Each set contains images from related classes. Our method captures regions of all common objects from different classes, contrary to supervised methods [31, 56]. Saliency Baseline [3] results are noisy.

dataset has forty sets of six images, each from semantically related classes (e.g., car-bus-train, bird-plane). Fig. 9 shows a sample set, the rest is in the SM.

Quantitative evaluation. We compare our *unsupervised* approach to state-of-the-art *supervised* methods, trained on large datasets with ground truth seg-

mentation masks, [56, 31, 29, 30]; and *unsupervised* methods, [15, 44, 54, 22, 28]. We report Jaccard Index ($\mathcal{J}_m$), which reflects both precision (covering the foreground) and accuracy (no foreground "leakage"), and mean precision ($\mathcal{P}_m$). The results appear in Table 2. 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.

Ablation. We conduct an ablation study to validate our observations in 3.1. As mentioned in Sec. 2, Caron et al. [3] observed DINO-ViT attention heads attend to salient regions in the image, and threshold them to perform object segmentation. We name their method "DINO-ViT Saliency Baseline" as mentioned in Table 3. We apply the same baseline with attention heads from a supervised ViT (Sup. ViT Saliency Baseline). To compare ViT with CNN representations, we also apply a similar method thresholding ResNet features (DINO / Sup. ResNet Saliency Baseline), implementation details are in the SM. We also ablate our method with different facets. The supervised ViT performs poorest, while both ResNet baselines perform similarly. The DINO-ViT baseline exceeds them and is closer to our performance. The remaining performance gap between our method and the DINO-ViT baseline can be attributed to one bias in the DINO-ViT baseline - it captures foreground salient objects regardless of their commonality to the other objects in the images. For example, the house behind the blue car in Fig. 9 is captured by DINO-ViT Saliency Baseline but is not captured by our method. This corroborates our observation that the properties of DINO-ViT stem from both architecture and training method. The facet ablation demonstrates our observation that keys are superior than other facets.

5.3 Point Correspondences

Qualitative Results. We test our method on numerous pairs, compared with the VGG-based method, NBB [1]. Figure 10 shows our results are more robust to changes of appearance, pose and scale on both intra- and inter-class pairs.

Quantitative Evaluation. We evaluate on 360 random Spair71k [36] pairs, and measure performance by 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 modify our method to match this evaluation protocol to compute the binned descriptors for the given keypoints in the source image, and find their nearest-neighbors in the target image. We compare to NBB[1] (VGG19-based) and CATs [7] (ResNet101-based). Table 4 shows that our method outperforms NBB *by a large margin*, and closes the gap towards the *supervised* CATs [7].

Ablations. We ablate our method using on different facets and layers, with and without binning. Table 4 empirically corroborates our observation of keys being a better than other facets, and that features from earlier layers are more sensitive to their position in the image. The correspondence task benefits from these intermediate features more than plainly using the deepest features (Sec. 3.1).

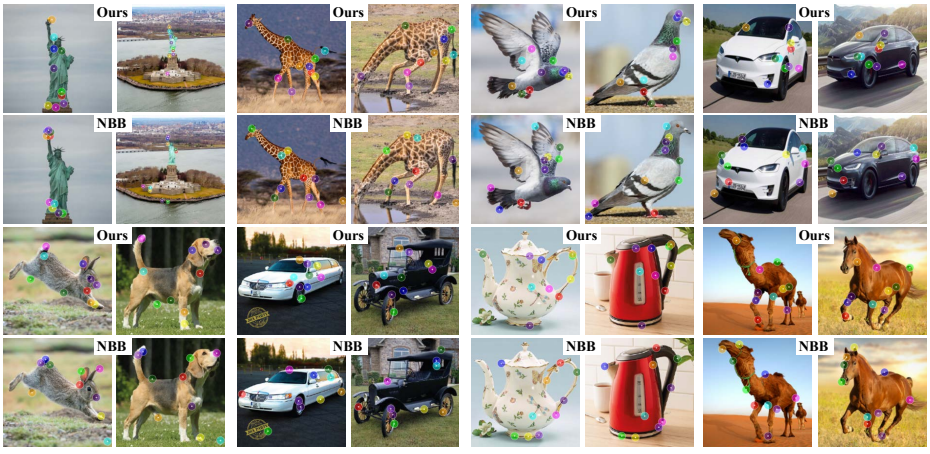

Fig. 10. *Correspondences Comparison to NBB [1]:* On intra-class (top-row) and inter-class (bottom-row) scenarios. Our method is more robust to appearance, pose and scale variations. Full size results are available in the SM.

Method	Layer 9				Layer 11				NBB [1]	Supervised [7]
	key	query	value	token	key	query	value	token		
with bins	<u>56.48</u>	54.96	52.33	56.03	53.45	52.35	49.37	50.34	26.98	**61.43**
without bins	52.27	49.35	43.97	50.14	47.08	42.64	41.56	46.09		

Table 4. *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 [7] for reference.

6 Conclusion

We provided new empirical observations on the internal features learned by ViTs under different supervisions, and harnessed them for several real-world vision tasks. We demonstrated the power of these observations by applying only lightweight zero-shot methodologies to these features, and still achieving competitive results to state-of-the-art supervised methods. We also presented new capabilities of part co-segmentation across classes, and on domains that lack available training sets. We believe that our results hold great promise for considering deep ViT features as an alternative to deep CNN features.

Acknowledgments: We thank Miki Rubinstein, Meirav Galun, Kfir Aberman and Niv Haim for their insightful comments and discussion. This project received funding from the Israeli Science Foundation (grant 2303/20), and the Carolito Stiftung. Dr Bagon is a Robin Chemers Neustein Artificial Intelligence Fellow.

References

1. Aberman, K., Liao, J., Shi, M., Lischinski, D., Chen, B., Cohen-Or, D.: Neural best-buddies: Sparse cross-domain correspondence. TOG (2018)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. ECCV (2020)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. ICCV (2021)
4. Carter, S., Armstrong, Z., Schubert, L., Johnson, I., Olah, C.: Activation atlas. Distill (2019)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence (2017)
6. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. CVPR (2014)
7. Cho, S., Hong, S., Jeon, S., Lee, Y., Sohn, K., Kim, S.: Semantic correspondence with transformers. NeurIPS (2021)
8. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. CVPR (2020)
9. Choudhury, S., Laina, I., Rupprecht, C., Vedaldi, A.: Unsupervised part discovery from contrastive reconstruction. NeurIPS (2021)
10. Collins, E., Achanta, R., Susstrunk, S.: Deep feature factorization for concept discovery. In: The European Conference on Computer Vision (ECCV) (2018)
11. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. ICLR (2019)
12. Dekel, T., Oron, S., Rubinstein, M., Avidan, S., Freeman, W.T.: Best-buddies similarity for robust template matching. CVPR (2015)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
14. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV (2015)
15. Faktor, A., Irani, M.: Co-segmentation by composition. ICCV (2013)
16. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the nyström method. TPAMI (2004)
17. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. CVPR (2016)
18. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. ICLR (2019)
19. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR (2014)
20. Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T.: Unsupervised semantic segmentation by distilling feature correspondences. ICLR (2022)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CVPR (2016)

22. Hsu, K.J., Lin, Y.Y., Chuang, Y.Y.: Co-attention cnns for unsupervised object co-segmentation. IJCAI (2018)
23. Huang, Z., Li, Y.: Interpretable and accurate fine-grained recognition via region grouping. CVPR (2020)
24. Hung, W.C., Jampani, V., Liu, S., Molchanov, P., Yang, M.H., Kautz, J.: Scops: Self-supervised co-part segmentation. CVPR (2019)
25. Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., Yi, K.M.: COTR: correspondence transformer for matching across images. ICCV (2021)
26. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. ECCV (2016)
27. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. NeurIPS (2011)
28. Kuang-Jui Hsu, Yen-Yu Lin, Y.Y.C.: Deepco3: Deep instance co-segmentation by co-peak search and co-saliency detection. CVPR (2019)
29. Li, B., Sun, Z., Li, Q., Wu, Y., Hu, A.: Group-wise deep object co-segmentation with co-attention recurrent neural network. ICCV (2019)
30. Li, G., Zhang, C., Lin, G.: Cyclesegnet: Object co-segmentation with cycle refinement and region correspondence. TIP (2021)
31. Li, W., Jafari, O.H., Rother, C.: Deep object co-segmentation. ACCV (2018)
32. Liu, S., Zhang, L., Yang, X., Su, H., Zhu, J.: Unsupervised part segmentation through disentangling appearance and shape. CVPR (2021)
33. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. ICCV (2015)
34. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. NeurIPS (2016)
35. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. ECCV (2018)
36. Min, J., Lee, J., Ponce, J., Cho, M.: Spair-71k: A large-scale benchmark for semantic correspondence. CoRR (2019)
37. Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Intriguing properties of vision transformers. NeurIPS (2021)
38. Ng, A.: Clustering with the k-means algorithm. Machine Learning (2012)
39. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. NeurIPS (2001)
40. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017)
41. Poličar, P.G., Stražar, M., Zupan, B.: opentsne: a modular python library for t-sne dimensionality reduction and embedding. bioRxiv (2019)
42. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? NeurIPS (2021)
43. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. TOG (2004)
44. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. CVPR (2013)
45. Rubio, J.C., Serrat, J., López, A., Paragios, N.: Unsupervised co-segmentation through region matching. CVPR (2012)
46. Shocher, A., Gandelsman, Y., Mosseri, I., Yarom, M., Irani, M., Freeman, W.T., Dekel, T.: Semantic pyramid for image generation. CVPR (2020)
47. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. ECCV (2006)

48. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., Ponce, J.: Localizing objects with self-supervised transformers and no labels. BMVC (2021)
49. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)
50. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. CVPR (2021)
51. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks by factorized spatial embeddings. ICCV (2017)
52. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. ICLR (2022)
53. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. CVPR (2018)
54. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR (2022)
55. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
56. Zhang, K., Chen, J., Liu, B., Liu, Q.: Deep object co-segmentation via spatial-semantic network modulation. AAAI (2020)
57. Zhang, R.: Making convolutional networks shift-invariant again. ICML (2019)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. CVPR (2018)
59. Zhang, Y., Guo, Y., Jin, Y., Luo, Y., He, Z., Lee, H.: Unsupervised discovery of object landmarks as structural representations. CVPR (2018)