Self-Supervised Learning of Object Parts for Semantic Segmentation

Adrian Ziegler
Technical University of Munich
adrian.ziegler@tum.de

Yuki M. Asano
QUVA Lab
University of Amsterdam
y.m.asano@uva.nl

Abstract

Progress in self-supervised learning has brought strong image representation learning methods. Yet so far, it has mostly focused on image-level learning. In turn, tasks such as unsupervised image segmentation have not benefited from this trend as they require spatially-diverse representations. However, learning dense representations is challenging, as in the unsupervised context it is not clear how to guide the model to learn representations that correspond to various potential object categories. In this paper, we argue that self-supervised learning of object parts is a solution to this issue. Object parts are generalizable: they are a priori independent of an object definition, but can be grouped to form objects a posteriori. To this end, we leverage the recently proposed Vision Transformer's capability of attending to objects and combine it with a spatially dense clustering task for fine-tuning the spatial tokens. Our method surpasses the state-of-the-art on three semantic segmentation benchmarks by 17%-3%, showing that our representations are versatile under various object definitions. Finally, we extend this to fully unsupervised segmentation – which refrains completely from using label information even at test-time – and demonstrate that a simple method for automatically merging discovered object parts based on community detection yields substantial gains. .

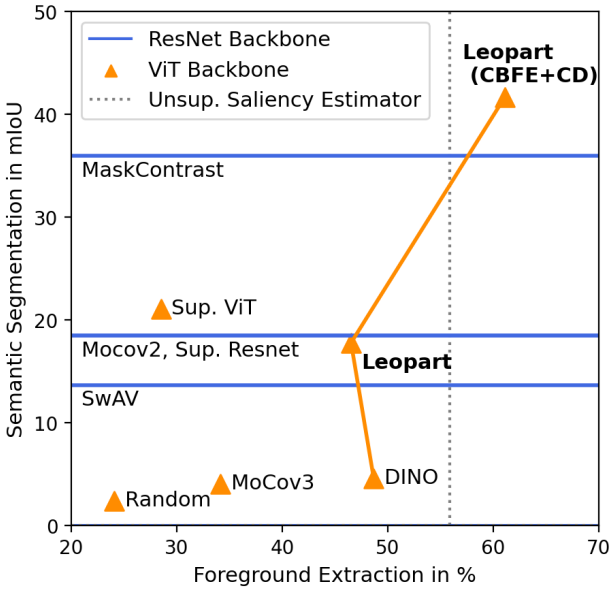


Figure 1. **ViTs and Resnets compared under foreground extraction and semantic segmentation.** We use Jaccard distance as a measure for foreground extraction. Starting from a DINO initialization, our method, Leopart, closes the performance gap between self-supervised ViTs and their supervised counterparts as well as Resnets. Leopart (CBFE+CD) further improves a ViT's object extraction capabilities and sets new state-of-the-art for fully unsupervised semantic segmentation.

1. Introduction

Defining what makes an object an object is hard. In philosophy, Peirce defines an object as anything we can think and talk about [40]. In computer vision, object definitions for semantic segmentation are more pragmatic and feature various notions of objectness as well as different levels of granularity. For instance, the COCO-Stuff benchmark distinguishes between stuff (objects without a clear shape) and things (objects with a "well-defined" shape) [3, 35] and features coarse and fine object categories. Others, like Cityscapes [12], choose a segmentation that is most informative for a specific application like autonomous driving and therefore also include sky and road as object classes.

This variedness of object definitions is challenging for self-supervised or unsupervised semantic segmentation as human annotations that carry the object definitions are, at most, used at test time. However, the ability to learn self-supervised dense representations is desirable as this would allow scaling beyond object-centric images and allow effective learning on billions more generic images. Furthermore, unsupervised segmentation can be highly useful as a starting point for more efficient data labeling, as segmen-

Code: https://github.com/MkuuWaUjinga/leopart

tation annotations are even more expensive than image labelling [35]. To tackle the lack of a principled object definition during training, many methods resort to defining object priors such as saliency and contour detectors to induce a notion of objectness into their pretext tasks [9, 28, 30, 49, 56], effectively rendering such methods semi-supervised and potentially not generalizeable. In this paper, we instead stay in the fully unsupervised domain and explore a novel yet simple alternative for training densely. We **le**arn **o**bject **part**s (Leopart) through a dense image patch clustering pretext task. Object part learning promises a principled formulation for self-supervised dense representation learning as object parts can be composed to form objects as defined in each benchmark, after generic pretraining.

In this paper, we explore the use of a Vision Transformer (ViT) with our new loss and excavate its unique aptness for self-supervised segmentation. While vision transformers have shown great potential unifying architectures and scaling well with data into billions, they have mostly been shown to work for image-level tasks [6, 8, 16] or dense tasks [11, 37, 50, 52] but in a supervised manner. Our work aims to close this gap by self-supervisedly learning dense ViT models. We combine the recently discovered property of self-supervised ViTs to localise objects [6] with our dense loss to train spatial tokens for unsupervised segmentation.

We validate our method from two different angles: First, we conduct a transferability study and show that our representations perform well on downstream semantic segmentation tasks. Second, we tackle the more challenging *fully* unsupervised setup proposed in [49] based on directly clustering the pixel or patch embeddings. For that, two model characteristics are important: unsupervised foreground extraction and a semantically-structured embedding space, see Figure 1. To our surprise, even though self-supervised ViTs excel at extracting objects, they do not learn a spatial token embedding space that is discriminative for different object categories. On the other hand, ViTs trained under supervision achieve better semantic segmentation performance, but the attention heads perform poorly at localizing objects. In contrast, our method outperforms self-supervised ViTs *and* Resnets in fully unsupervised semantic segmentation as well as in learning transferable dense representations.

Thus, our contributions are as follows:

- We propose a dense clustering pretext task to learn semantically-rich spatial tokens closing the gap between supervised ViTs and self-supervised ViTs.
- We show that a ViT trained with our pretext task learns transferable representations that surpass the state-of-the-art on Pascal VOC, COCO-Thing and COCO-Stuff semantic segmentation *at the same time* by 17%-3%.
- We develop a novel cluster-based foreground extraction and overclustering technique based on community detection to tackle fully unsupervised semantic segmentation and surpass the state-of-the-art by >3%.

2. Related Work

Our work takes inspiration from standard image-level self-supervised pretraining while extending this to the domain of dense representation learning using Vision Transformers.

Image-level self-supervised learning. Self-supervised learning aims to learn powerful representations by replacing human annotation with proxy tasks derived from data alone. Current methods can be roughly categorized into instance-level and group-level objectives. Instance-level objectives include predicting augmentations applied to an image [15, 22, 31, 38, 39, 54, 55] or learning to discriminate between images [2, 7, 17, 25, 53], often done by the use of contrastive losses [24].

On the other hand, group-level objectives explicitly allow learning shared concepts between images by leveraging clustering losses [1, 4, 5, 48]. [4] proposes k-means clustering in feature space to produce pseudo labels for training a neural network. [1] casts the problem of finding pseudo labels as an optimal transport problem unifying the clustering and representation learning objectives. This formulation was adapted to an online setting in SwAV [5] together with a new multi-crop augmentation strategy, a random cropping method that distinguishes between global and local crops of an image. The IIC method [32], also conducts clustering, however using a mutual information objective. While it can also be used densely, it has been found to focus on lower-level features specific to each dataset [49]. Another recent line of works completely refrains from group level clustering or instance-based discrimination by predicting targets from a slowly moving teacher network [6, 23].

Our work adapts this teacher-student setup and shows its benefits beyond image-level tasks. To this effect, we build on the clustering pretext task from [5] and reformulate it such that it can be used on an image patch level with teacher-student setups. We also use the multi-crop augmentation strategy and provide an interpretation from the perspective of dense prediction tasks.

Dense self-supervised learning. Based on the observation that image-level learning does not imply expressive dense representations [26, 41], dedicated self-supervised dense representation learning has attracted a lot of attention [10, 20, 21, 28–30, 32, 34, 36, 44, 46, 49, 51, 57]. DenseCL reformulates the contrastive objective used in MoCo [25] to work on spatial features by establishing dense correspondences accross views and is currently the state-of-the-art in transfer learning for semantic segmentation on PVOC [34]. Other methods resort to defining an unsupervised object prior such as region proposals [9], contour de-

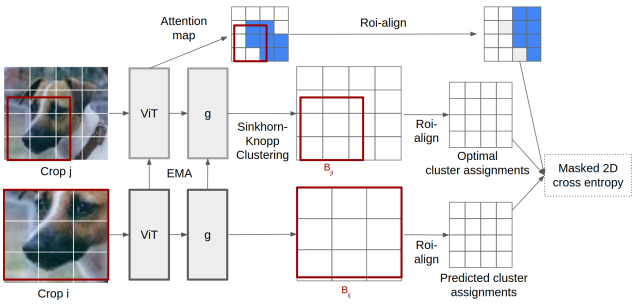


Figure 2. **Leopart training pipeline.** We start from a DINO initialization. We feed different crops to the student and teacher network to produce patch-level cluster predictions and optimal cluster assignments targets. This requires an alignment step of cluster targets and assignments. We further focus clustering on foreground patches by leveraging the ViT's attention map.

tectors [30, 57], saliency [49] or object masks [28]. For instance, the current state-of-the-art for unsupervised semantic segmentation, MaskContrast [49], uses a pretrained saliency estimator to mine positive and negative pixels for contrastive learning.

Concurrent to our work, [34] proposes an intra-image clustering step of pixel embeddings before applying a contrastive loss on the identified pixel groups to segment images. However, as they are reliant on combining the former with an image-level loss, it is not well-suited for more generic images with multiple objects where image-level and pixel-level semantics do not match. In contrast, our method uses a single clustering objective that is generalized for the dense setting, but that also works on object-centric images. Furthermore, by leveraging ViT's natural ability to direct its attention to objects, we do not require any external saliency generator like [49].

There are also works that have explored unsupervised object parts segmentation [10, 29], with the explicit goal to determine parts given object masks. However, our goal is different as we use part representations as an intermediary *for* semantic segmentation on classic, object-level settings.

Superficially similar to our work is also another concurrent work [46], which tackles object *detection* by using the similarity between DINO's frozen last layer self-attention patch keys as a metric for merging image patches to objects. In contrast, we use DINO's *spatial tokens* and propose to fine-tune them for *semantic segmentation*.

3. Method

Our goal is to learn an embedding space that groups image patches containing the same part of an object.

This is motivated by the hypothesis that object part representations are more general than object representations, as parts can be recombined in multiple ways to different objects. As an example, a wheel representation can be combined to a car representation but also a bus representation. Therefore, object part representations should transfer better across datasets. For that, we aim to design a pretext task that allows for intra-image category learning on an image-patch-level. Thus, a clustering pretext task is a natural choice. As shown in Figure 2, we retrieve patch-level optimal cluster assignments from a teacher network and predict them from the student network. The choice of a clustering pretext task is further supported by empirical evidence showing that clustering pretext outperforms contrastive pretext for dense prediction tasks [20,34]. Instead of pretraining models from scratch which requires substantial GPU budgets, we use our loss to fine-tune pretrained neural networks. Further, this circumvents known cluster stability issues and clusters capturing low-level image features when applied to a patch-level as reported in [48].

3.1. Fine-tuning loss for spatial tokens

Image Encoder. Given an image $x \in \mathbb{R}^{3\times H\times W}$, we flatten the image into $N = \lfloor \frac{H}{P} \rfloor \cdot \lfloor \frac{W}{P} \rfloor$ separate patches $x_i, i \in 1, \dots, N$ of size $P \times P$ each. The vision encoder we use is a ViT [16], which at the maps the image patches x_i to a vector of N spatial tokens $f(x) = [f(x_1), \dots f(x_N)]$.

Leopart fine-tuning loss. To train the ViT's spatial tokens, we first randomly crop the image V-times into v_g global views and v_l local views. When sampling the views we compute their pairwise intersection in bounding box format and store it in a matrix B. We denote the transformed version of the image as $x_{t_j}, j \in \{1, \dots, V\}$. Then, we forward the spatial tokens through a MLP projection head g with a L2-normalization bottleneck to get spatial features for each crop: $g(f(x_{t_j})) = Z_{t_j} \in \mathbb{R}^{D\times N}$. To create prediction targets, we next find an optimal soft cluster assignment Q_{t_j} of all spatial token's feature vector Z_{t_j} to K prototype vectors $[c_1, \dots, c_K] = C \in \mathbb{R}^{D\times K}$. For that, we follow the online optimization objective of SwAV [5] that works on the whole image batch b. Q is optimized such that the similarity between all feature vectors in the batch and the prototypes is maximized, while at the same time being regularized towards assigning equal probability mass to each prototype vector. This can be cast to an optimal transport problem and is solved efficiently with the Sinkhorn-Knopp algorithm [14]. Instead of optimizing over $|b|$ feature vectors, we instead optimize over $N \cdot |b|$ spatial feature vectors as we have N spatial tokens for each image. As our batchsizes are small, we utilize a small queue that keeps the past 8192 features, as is done in SwAV.

With the optimal cluster assignment of all image crops' spatial tokens $Q_{t_k} \in \mathbb{R}^{N\times K}$, we formulate a swapped prediction task:

$$L(x_{t_1}, ..., x_{t_V}) = \sum_{j=0}^{v_g} \sum_{i=0}^{V} \mathbb{1}_{k\neq j} l(x_{t_i}, x_{t_j}) \qquad (1)$$

Here, l is the 2D cross entropy between the softmaxed and aligned cluster assignment predictions and the aligned optimal cluster assignments:

$$l(x_{t_i}, x_{t_j}) = H\big[(s_\tau(\alpha_{B_{j,i}}(g(\Phi(x_{t_i}))^T C), \alpha_{B_{ij}}(Q_{t_j})\big], \tag{2}$$

where H is cross-entropy and s_τ a softmax scaled by temperature τ. We use L to jointly minimize the prototypes C as well as the neural networks f and g. C is further L2-normalized after each gradient step such that $Z^T C$ directly computes the cosine similarity between spatial features and prototypes.

Since global crops capture the majority of an image, we solely use these to compute Q_{t_j}, as the spatial tokens can attend to global scene information such that the overall prediction target quality improves. Further, using local crops to produce cluster assignment predictions effectively enables, as local crops just cover parts of images and thus also parts of objects, object parts to object category reasoning, an important ability for scene understanding.

Alignment. In Equation 2 we introduce the alignment operator $\alpha_{B_{ij}}(\cdot)$. This is necessary, because x_{t_j} and x_{t_i} cover different parts of the input image and thus Q_{t_j} and the cluster assignment prediction $Z_{t_i}^T C$ correspond to different image areas. To tackle this, $\alpha(\cdot)$ restores the spatial dimensions $\lfloor \frac{H}{P} \rfloor \times \lfloor \frac{W}{P} \rfloor$ and aligns the tensor using the crop intersection bounding boxes B_{ji} and B_{ij} respectively. In our experiments we use RoI-Align [27] as operator, producing features with a fixed and compatible output size.

Foreground-focused clustering. To focus the clustering on the foreground tokens, we further leverage the ViT's `CLS` token attention maps $A_i \in [0,1]^N$ of each of its attention heads. To create a foreground clustering mask that can be used during training, we first average the attention heads to one map and apply a Gaussian filter for smoothing. We then obtain a binary mask A_b by thresholding the map to keep 60% of the mass following [6]. We use $\alpha_{B_{ji}}$ to align the global crop's attention to the intersection with crop j. The resulting mask is then applied as 0-1 weighting to the 2D cross entropy loss, $l \odot A_b$. Note that we extract the attention maps and spatial tokens with the same forward pass, thus not impacting training speed.

3.2. Fully unsupervised semantic segmentation

In this section we describe our method that enables us to do fully unsupervised semantic segmentation. All of them work directly in the learned spatial token embedding space and are based on simple K-means clustering.

3.2.1 Cluster-based Foreground Extraction (CBFE)

Under the hypothesis that clusters in our learned embedding space correspond to object parts, we should be able to extract foreground objects by assigning each cluster id to foreground object (fg) or background (bg): $\Theta : \{1, \dots, K\} \rightarrow \{\text{fg}, \text{bg}\}$. Thus, at evaluation time, we construct Θ without supervision, by using ViT's merged attention maps A_b as a noisy foreground hint. Similar to how we process the attention maps to focus our clustering pretext on foreground, we average the attention heads, apply Gaussian filtering with a 7x7 kernel size and keep 60% of the mass to obtain a binary mask. Using train data, we rank all clusters by pixel-wise precision with A_b and find a good threshold c for classifying a cluster as foreground. This gives us Θ that we apply to the patch-level clustering to get a foreground mask.

3.2.2 Overclustering with community detection (CD)

As we will see from Table 1, the segmentation results improve substantially with higher clustering granularities. However, this is mainly because overclustering draws on label information to group clusters to ground-truth objects and in the limit of providing one cluster for each pixel, it would be equivalent to providing full supervision signals. Here, we propose a novel overclustering method that requires no supervision at all.

Assuming that our clusters correspond to object parts, we interpret objects as a set of object parts that co-occur together. Thus, local co-occurrence of clusters in an image should provide a hint about such objects. Using co-occurrence statistics to categorize objects has been proposed before in [19, 42]. However, we are the first to work with object parts and no labels and employ a novel network science method to discover objects. To group the clusters, we construct an undirected and weighted network $G = (V, E, w)$, with $v_i, i \in \{1, \dots, K\}$ corresponding to each cluster. Then, we calculate the conditional co-occurrence probability $P(v_j|v_i)$ for clusters i and j over all images D. We use a localized co-occurrence variant that regards the 8-neighborhood up to a pixel distance d. With the co-occurrence probabilities at hand, we define $w(e_{i,j}) = \min(P(v_j|v_i), P(v_j|v_i))$. This asymmetric edge weight definition is motivated by the fact that parts need not be mutually predictive: For instance, a car windshield might co-occur significantly with sky but presence of a sky is not predictive for a car windshield.

To find communities in G, we use the common Infomap algorithm [45] as it works with weighted graphs and scales linearly with $|E|$. It works by leveraging an information-theoretic definition of network communities: Random walks sample information flow in networks and inform the construction a map Θ_K from nodes to M communities minimizing the expected description length of a random walk. With the discrete many-to-one mapping $\Theta_K : V \rightarrow \{1, \dots, M\}$ obtained from Infomap and computed on train data, we merge the clusters of the validation data to

the desired number of ground-truth classes and do Hungarian matching [33]. Note that Hungarian matching does not extract any meaningful label information; it merely makes the evaluation metric permutation-invariant [32].

4. Experiments

In this section, we evaluate the image patch representations learned by Leopart. We first ablate design decisions of our method to find an optimal configuration. To evaluate whether some datasets are more information-rich for object parts learning than others, we also ablate training on different datasets. We further test the performance of our dense representations under a transfer learning setup for semantic segmentation. Furthermore, we show that Leopart can also be used for fully unsupervised segmentation requiring no label information at all for evaluation.

4.1. Setup

Evaluation protocols. For all experiments, we discard the projection head used during training. Instead we directly evaluate the ViT's spatial tokens. We use two main techniques for evaluation: linear classifier and overclustering. For linear classifier (LC), we fine-tune a 1x1 convolutional layer on top of the frozen spatial token or the pre-GAP `layer4` features, following [49]. For overclustering, we run K-Means on all spatial tokens of a given dataset. We then group cluster to ground-truth classes by greedily matching by pixel-wise precision and run Hungarian matching [33] on the merged cluster maps to make our evaluation metric permutation-invariant following [32]. We always report overclustering results averaged over five different seeds. Overclustering is of special interest as it works directly in the learned embedding space and therefore requires less supervision than training a linear classifier. For completeness we also report results fine-tuning a deeper fully-convolutional net (FCN) following [51]. Generally, we follow the fine-tuning procedures of prior works [49, 51, 57]. We report results in mean Intersection over Union (mIoU) unless specified otherwise.

Model training. We train a ViT-Small with patch size 16 and start training from DINO weights [6]. All models were trained for 50 epochs using batches of size 32 on 2 GPUs. Further training details are provided in the Appendix.

Datasets. We train our model on ImageNet-100, comprising 100 randomly sampled ImageNet classes [47], COCO [35] and Pascal VOC (PVOC) [18]. When fine-tuning on COCO-Stuff and COCO-Thing, we use a 10% split of the training sets. Evaluation results are computed on the full COCO validation data for COCO-Stuff and COCO-Thing and PVOC12 *val*. This setup up makes sure that our representations are assessed under varying object definitions (*e.g.* stuff vs. thing) and granularities. Further details are provided in the Appendix.

		Num. clusters		
mask	LC	100	300	500
all	67.4	37.9	44.6	47.8
bg	64.7	28.1	39.0	41.4
fg	**67.8**	**38.2**	**47.2**	**50.7**

(a) Focusing clustering on foreground (fg) helps.

		Num. clusters		
crops	LC	100	300	500
[2]	66.1	33.0	42.5	45.0
[2,2]	67.7	37.8	45.4	49.3
[2,4]	**67.8**	**38.2**	**47.2**	**50.7**

(b) Local crops boost performance.

		Num. clusters		
tchr	LC	100	300	500
✗	67.6	34.6	44.3	47.9
✓	**67.8**	**38.2**	**47.2**	**50.7**

(c) Using an EMA teacher helps.

		Num. clusters		
protos	LC	100	300	500
100	67.7	36.8	45.4	49.2
300	**67.8**	**38.2**	**47.2**	**50.7**
500	67.4	35.8	44.8	49.1

(d) 300 prototypes work well.

Table 1. **Ablations** of different design decisions for Leopart.

Dataset	size	LC	K=500	K=300	K=100
IN-100	126k	67.8	50.7	47.2	38.2
COCO	118k	**69.1**	**53.0**	**49.9**	**44.3**
PASCAL	10k	64.5	50.7	47.8	38.2

Table 2. **Training data study for Leopart**. We use the best performing model config from Table 1 and train on different datasets.

4.2. Fine-Tuning Loss Ablations

In this section, we ablate the most important design decisions and hyperparameters of our fine-tuning loss as well as the aptness of different datasets for learning object parts. We evaluate on PVOC *val* and report three different overclustering granularities next to LC results.

Model Ablation. In Table 1 we report the model ablation results. As described in Section 3, we propose to leverage attention maps to guide our clustering algorithm. Note that the attention maps are just a noisy approximation of foreground objects. Thus, it only *focuses* spatial token clustering on foreground but does not neglect objects in the background. We find that foreground clustering gives substantial performance gains over our two ablated versions: clustering of all spatial tokens (up to 3%) and clustering mostly background tokens (up to 10%), as shown in Table 1a.

In Table 1b we ablate the multi-crop augmentation strategy. More specifically, we compare using four local crops against using only two or no local crops. The usage of local crops (last vs. second to last row) gives a much larger performance gain than using just more local crops (second to last vs. first row). This shows that predicting cluster assignments from constrained local image information is an important aspect for learning expressive dense representations. Interestingly, the overclustering results are effected more by this ablation, showing that local crops are important for learning a semantically-structured embedding space.

We also ablate the number of prototypes used for Sinkhorn-Knopp clustering in Table 1d. We find that the best performance is achieved with a moderate overcluster-

Method	Train	PVOC12		COCO-Things		COCO-Stuff	
		LC	K=500	LC	K=500	LC	K=500
Sup. ViT	IN + IN21	68.1	55.1	65.2	50.9	49.0	35.1
Sup. ResNet	IN	53.8	36.5	57.8	44.2	44.4	30.8
instance-level:							
MoCo-v2 [25]	IN	45.0†	39.1	47.5	36.2	32.6	28.3
DINO [6]	IN	50.6	17.4	50.6	23.5	47.7	32.1
SwAV [5]	IN	50.7†	35.7	56.7	37.3	46.0	33.1
pixel/patch-level:							
IIC [32]	PVOC	28.0†	-	-	-	-	-
MaskContrast [49]	IN+PVOC	49.2	45.4	47.5	37.0	32.0	25.6
DenseCL [51]	IN	49.0	43.6	53.0	41.0	40.9	30.3
Leopart	IN	68.0	50.5	62.5	49.2	51.2	**43.8**
Leopart	IN+CC	**69.3**	**53.3**	**67.6**	**55.9**	**53.5**	43.6

Table 3. **Transfer learning for semantic segmentation results.** Best results are in **bold** and second best are underlined. 'IN', 'IN21', 'CC' and 'PVOC' indicate training on ImageNet, ImageNet21k, CoCo and Pascal *trainaug* respectively. † indicates result taken from [49].

Method	mIoU
Sup. ResNet	18.5
Sup. ViT	21.1
DINO [6]	4.6
SwAV [25]	13.7
MoCo-v2 [25]	18.5
MaskContrast [49]	35.0†
Leopart (CBFE+CD)	**41.7**

Table 4. **Unsupervised semantic segmentation results.** We outperform other state-of-the-art methods by a large margin. † indicates result taken from [49].

	mIoU
K=150	48.8
DINO	4.6
+ Leopart	18.9 (+14.3%)
+ CBFE	36.6 (+17.7%)
+ CD	41.7 (+5.1%)

Table 5. **Component contributions.** We show the gains that each individual component brings for PVOC segmentation and K=21.

ing of 300 prototypes. Note however, that the number of prototypes we use for training is not equivalent to the number of clusters used for evaluation, which we denote for instance by K=500. Lastly, even though we fine-tune a pretrained model, we find that an EMA teacher still helps with learning more expressive representations as can be seen in Table 1c.

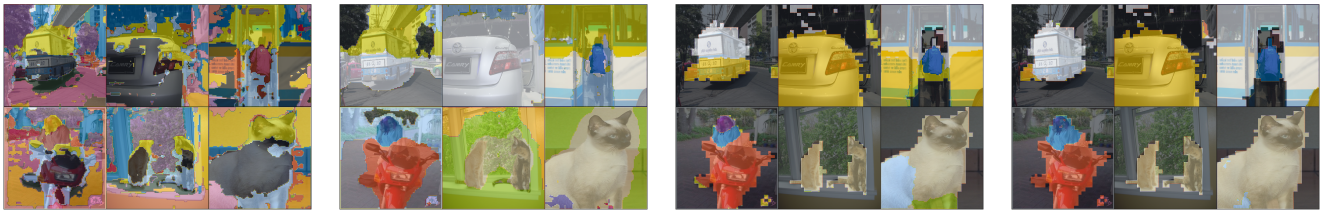
Training Data. In Table 2 we report results under varying training data: ImageNet-100, COCO and PVOC. ImageNet-100 usually features object-centric images with few objects. In comparison, COCO and PVOC contain images with more complex scenes. For comparability, we adapt the number of epochs for PVOC to 500 such that all models are trained for the same number of iterations. We find that our method's performance improves by up to 6% when trained on COCO instead of ImageNet-100. This shows the potential of our dense clustering loss when applied to less object-centric images and is in stark contrast to other methods reporting that their results get worse when training on COCO instead of ImageNet [34, 51, 57]. Finally, we see the worse results on PVOC as a confirmation of the fact that even for fine-tuning, larger data sets perform better for ViTs.

4.3. Transfer learning

Next, we study how well our dense representations, once learned, generalize to other datasets. We train our model on ImageNet-100 or COCO and report LC and overclustering results on PVOC12, COCO-Things and COCO-Stuff. As shown in Table 3, we outperform self-supervised prior works by large margins *on all three datasets* even though some use further datasets and supervision. On PVOC12 we surpass the state-of-the-art by more than 17% for linear evaluation and by more than 5% for overclustering. On COCO-Things and COCO-Stuff we improve linear classifier by $> 5\%$ and $> 3\%$ and overclustering by $> 8\%$ and $> 10\%$ respectively. Note that these gains are not due to the DINO initialisation nor due to ViTs per-se as the starting DINO model performs on par with other instance-level self-supervised methods that use ResNets like SwAV. In fact, DINO's embedding space exhibits inferior semantic structure in comparison to MoCo-v2 and SwAV as can be seen from the overclustering results on PVOC12 (-18%) and COCO-Things (-12%). Our method is also on par with the performance of a supervised ViT even though it was trained on a >10x times larger full ImageNet (IN-21k) dataset [43]. When fine-tuning on COCO instead of IN-100, we see further improvements on all datasets. The results confirm that

Method	Train	PVOC12 FCN
SegSort [30]	CC+PVOC	36.2†
Hier. Grouping [56]	CC+PVOC	48.8†
DINO [6]	IN	60.6
Hier. Grouping [56]	IN	64.7†
MoCo-v2 [25]	CC	64.5†
MoCo-v2 [25]	IN	67.5†
DenseCL [51]	CC	67.5†
DenseCL [51]	IN	69.4†
Leopart	IN	70.1
Leopart	IN+CC	**71.4**
Leopart (ViT-B/8)	IN+CC	**76.3**

Table 6. **FCN transfer learning results**. We follow the same notation as in Table 3. Note that Hierarchical Grouping and Segsort fine-tune a larger ASPP decoder. † indicates result taken from [51, 56]



(a) DINO (b) + Leopart (c) + CBFE (d) + CD

Figure 3. **Qualitative Segmentations by DINO and our gradual improvements**. We cluster the spatial tokens and visualize the resulting clusters obtained after each step of our method.

it is desirable to learn object parts representations, as they work well under different object definitions, as evidenced by strong performances across datasets.

In Table 6, we evaluate Leopart by fine-tuning a full FCN on top of frozen features. Again, we outperform all prior works, including DenseCL, the current state-of-the-art. Interestingly, while DenseCL shows a performance gain of more than 20% when fine-tuning a FCN instead of a linear layer, our performance gain from fine-tuning is relatively low at around 2%. We hypothesize that this behaviour is because our learned embedding space is already close to maximally informative for semantic segmentation under linear transformations. In contrast, DenseCL's embedding space alone is less informative in itself and requires a more powerful non-linear transformation. We push state-of-the-art even further by fine-tuning a larger ViT-Base with patch size 8 (ViT-B/8) improving FCN performance by around 5%. We report further details and experiments in the Appendix.

4.4. Fully unsupervised semantic segmentation

Encouraged by our strong K=500 overclustering results in Table 3, we next evaluate fully unsupervised semantic segmentation. This relies only on the learned embedding space's structure and refrains from using any test-time label information, *i.e*. the number of final clusters needs to be equivalent to the ground-truth. To that extent, we start with a simple K-means clustering of the spatial tokens to get cluster assignments for each token. As prior works, we base our evaluation on PVOC12 *val* and train self-supervised on an arbitrary dataset [49], in this case COCO. In Table 4 we compare our method to prior state-of-the-art. We outperform our closest competitor, MaskContrast, by $> 4\%$. While like MaskConstrast, we cluster only foreground tokens, we use our embedding space clustering instead of a pretrained unsupervised saliency estimator to do cluster-based foreground extraction (CBFE). Also, instead of averaging the feature representations per image, we use our novel unsupervised overclustering method with community detection (CD), allowing us to detect multiple object categories in one image.

Method	Jacc. (%)	B-F1 [13] (%)
DINO attention [6]	48.7	36.5
Unsup. saliency [49]	55.9	<u>40.8</u>
Leopart IN CBFE	<u>58.6</u>	**42.1**
Leopart CC CBFE	**59.6**	40.7

Table 7. **Foreground extraction results on PVOC val**. Our method improves over DINO atenttions with respect to Jaccard distance and Boundary F1 score and shows performance on par with a dedicated unsupervised saliency estimator.

(a) Attention Masks (b) Cluster masks

Figure 4. **DINO Attention masks vs. Leopart Cluster masks**.

4.4.1 Performance gain study

In Figure 3, we show the gradual visual improvement of the segmentations. By using Leopart we substantially improve the DINO baseline by more than 14%, as shown quantitatively in Table 5. This is also apparent when comparing Figure 3a to Figure 3b. The DINO segmentations show no correspondence to object categories, whereas the segmentations obtained by Leopart assign the same colors to the bus in the first and third image of the top row as an example. However, our segmentations do not correspond well with PVOC's object definitions, as we oversegment background. To further improve this, we extract foreground resulting in the segmentation maps shown in Figure 3c. The segmentation focuses on the foreground and object categories start to emerge more visibly. However, some objects are still oversegemented such as busses and cats. Thus, we run our proposed community detection algorithm to do fully unsupervised overclustering, resulting in the segmentations shown in Figure 3d.

CBFE. For foreground extraction, we follow the method proposed in Section 3.2.1. As shown in Table 7, our foreground masks obtained through CBFE outperform DINO's

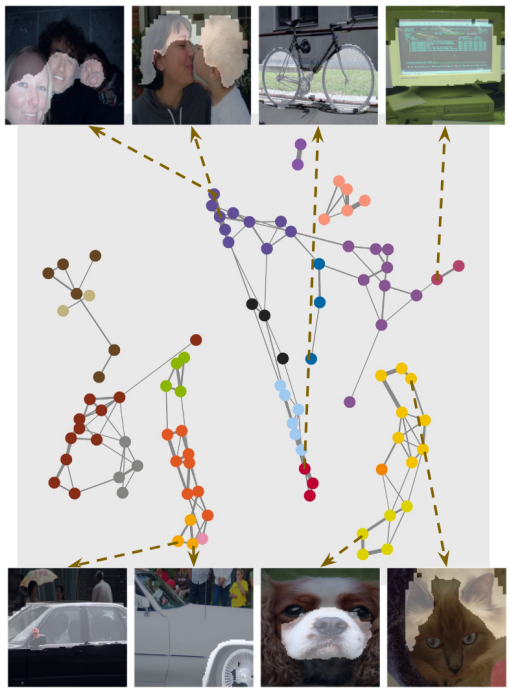


Figure 5. **Communities found in our cluster co-occurrence network constructed through self-supervision.** Each node corresponds to a cluster in our learnt embedding space. The nodes are colored by community membership.

attention maps by more than 9%. This is remarkable as we can only improve the attention map if the foreground clusters also segment the foreground correctly where the noisy foreground hint from DINO's attention is wrong. In Figure 4, we show mask visualizations to provide a qualitative idea of this phenomenon. While the attention masks only mark the most discriminative regions they fail to capture the foreground object's shape (Fig. 4(a)). Our cluster masks, however, alleviate this providing a crisp foreground object segmentation (Fig. 4(b)). With the foreground masks extracted, we can specify K-Means to run only on foreground spatial tokens. This further improves our fully unsupervised segmentation performance by $> 17\%$, as shown in Table 5.

CD. We have seen that overclustering yields benefits in terms of performance but requires additional supervision for merging clusters during evaluation. To reap the benefits of this process whilst staying unsupervised, we construct a network based on cluster co-occurrences and run community detection (CD) following the method proposed in Section 3.2.2. We find that CD can further improve our performance by $> 5\%$ and brings our fully unsupervised semantic segmentation results closer to the upper bound of supervised overclustering at test-time with $K = 150$, as shown in Table 5.

Finally, in Figure 5 we show a visualization of the constructed network, the discovered communities as well as some exemplary parts clusters. For instance, Leopart discovers bicycle wheels and car wheels separately. This demonstrates that we can learn high-level semantic clusters that do not latch on low-level information such as shape. Furthermore, we can observe that clusters that are semantically similar, such as human hair and human faces, are also part of the same community and close in the resulting network. Also, a gradual semantic transition within connected components can be observed as shown for dog snout and cat ears being part of different communities that are interconnected.

5. Discussion

Limitations. Since we do not learn on a pixel but on a patch level, our segmentation maps are limited in their resolution and detection capabilities. Thus, our method will fail when fine-grained pixel-level segmentation is required or very small objects covering less than an image patch are supposed to be segmented. Further, our unsupervised overclustering method does a hard assignment of clusters to communities. This has the limitation that object parts which occur in several objects are assigned to the wrong object category when they appear in a specific context. We show an example of this phenomenon in the Appendix, but leave a solution to this to future work.

Potential negative societal impact. Self-supervised semantic segmentation can scale to large datasets with little to no human labelling effort and extract information from it. However, as human input is kept to a minimum, rigorous monitoring of the segmentation results is mandatory, as objects might not be segmented in a way that we are used to or problematic biases in the data might become apparent. Lack of monitoring, could have potential negative impacts in areas such as autonomous driving and virtual reality.

Conclusion. In this paper, we propose a dense clustering pretext task for the spatial tokens of a ViT learning a semantically-rich embedding space in contrast to other self-supervised ViTs. We motivate our pretext task by observing that object definitions are brittle and object parts learning promises a principled alternative. Our experiments show that this formulation is favorable as we improve state-of-the-art on PVOC, COCO-Stuff and COCO-Thing semantic segmentation benchmarks featuring different object defintions and granularities. Our semantically-rich embedding space can also be directly used for fully unsupervised segmentation, showing that objects can be defined as co-occurring object parts.

Acknowledgements

A.Z. is thankful for using compute while interning at Pina Earth. Y.M.A is thankful for MLRA funding from AWS.

References

[1] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 2

[2] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise, 2017. 2

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 1

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 3, 6

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 4, 5, 6, 7

[7] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, and Heewoo Jun. Generative pretraining from pixels. In *ICML*, 2020. 2

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 2

[9] Minsu Cho, Suha Kwak, Ivan Laptev, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in images and videos. In *International conference on ubiquitous robots and ambient intelligence (URAI)*, pages 292–293. IEEE, 2015. 2

[10] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. In *NeurIPS*, volume 35, 2021. 2, 3

[11] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[13] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *Bmvc*, volume 27, pages 10–5244. Bristol, 2013. 7

[14] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 26:2292–2300, 2013. 3

[15] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[16] 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, 2021. 2, 3

[17] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks, 2015. 2

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[19] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8. IEEE, 2008. 4

[20] Shang-Hua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation, 2021. 2, 3

[21] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Perez. Obow: Online bag-of-visual-words generation for self-supervised learning. In *CVPR*, pages 6830–6840, 2021. 2

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018. 2

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020. 2

[24] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. 2, 6

[26] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training, 2018. 2

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 4

[28] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. *arXiv preprint arXiv:2103.10957*, 2021. 2, 3

[29] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, pages 869–878, 2019. 2, 3

[30] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *CVPR*, pages 7334–7344, 2019. 2, 3, 6

[31] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts, 2018. 2

[32] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 2, 5, 6

[33] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[34] Xiaoni Li, Yu Zhou, Yifei Zhang, Aoting Zhang, Wei Wang, Ning Jiang, Haiying Wu, and Weiping Wang. Dense semantic contrast for self-supervised visual representation learning. *arXiv preprint arXiv:2109.07756*, 2021. 2, 3, 6

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1, 2, 5

[36] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet, 2021. 2

[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 2

[38] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016. 2

[40] Charles Peirce. Reflections on real and unreal objects. 1

[41] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases, 2020. 2

[42] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 4

[43] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *NeurIPS-Data*, 2021. 6

[44] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *CVPR*, pages 1144–1153, 2021. 2

[45] Martin Rosvall and Carl T Bergstrom. Maps of information flow reveal community structure in complex networks. *arXiv preprint:0707.0609*, 2007. 4

[46] 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. *arXiv preprint arXiv:2109.14279*, 2021. 2, 3

[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 5

[48] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, pages 268–285, 2020. 2, 3

[49] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, 2021. 2, 3, 5, 6, 7

[50] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *ICCV*, 2021. 2

[51] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021. 2, 5, 6

[52] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *ICCV*, 2021. 2

[53] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2

[54] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data, 2019. 2

[55] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[56] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *arXiv preprint arXiv:2012.03044*, 2020. 2, 6

[57] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020. 2, 3, 5, 6