Appendix

A. Data augmentations for SSL

We use the following augmentations for the part contrast training step of our method. We list the details of each augmentation and the probability of applying it for each view x and x' which are passed to the main encoder and momentum encoder respectively.

Augmentation	p(x)	p(x')
Resized Cropping : Aspect ratio in [0.75,1.33], Area in [0.08,1.0]	1.0	
Horizontal Flip	0.5	
Color Jitter : Brightness in [0.6,1.4], Contrast in [0.6,1.4], Saturation in [0.8,1.2], Hue in [-0.1,0.1]	0.8	
Gaussian Blurring : Kernel size =23, Standard Deviation in [0.1,2.0]	1.0	0.1
Solarization : Threshold = 0.5	0.0	0.2

B. Comparison of Various Objectives for Fine-tuning

Tab. 4 compares the effect of training objectives from baselines and prior work. Fixing the initization to be MoCo V2 trained on ImageNet, we fine-tune on the CUB dataset using the objectives of MoCo V2, DetCon and ODIN. We list the performance on linear evaluation (cls.) and part segmentation (seg.). Even without the iterative process, our method outperforms all the baselines. We see a significant boost on iterating 3× especially in the performance on classification using linear evaluation. Improvement over vanilla DetCon can be attributed to the reliance of color and texture features which are less effective on birds due to their presence in cluttered backgrounds. Improvements over ODIN might be attributed to our reliance on hypercolumn representations instead of last-layer activations and a different learning objective.

Pre-training	**Cls.**	**Seg.**
ImageNet MoCo	28.92	46.08
MoCo	31.17	46.22
DetCon	32.00	44.58
ODIN	31.19	44.23
PARTICLE (1× iter.)	34.31	46.39
PARTICLE (3× iter.)	**36.09**	**47.40**

Table 4. **Comparison with standard baselines trained on the Caltech-UCSD birds dataset.** We present the performance gain obtained by PARTICLE compared with standard baselines (see § 5.3) when all are fine-tuned self-supervisedly on the CUB dataset. For all the fine-tuned methods we initialize weights with MoCo V2 trained on ImageNet. Again our method using clusters obtained from DetCon hypercolumns beats all baselines. The performance is especially boosted on Logistic Regression (Cls.) compared to ImageNet MoCo.

C. PiCIE Results

We train PiCIE [8] using their official code on the Caltech-UCSD birds dataset. We use the default hyperparameters with a ResNet50 and train for 20 epochs as used in the paper. We initialize the model with DetCon ImageNet instead of randomly so as to have a fair comparison with our method. We also set the number of clusters to be 25 same as our method. Fig 6 shows the segmentation maps produced after this training on a few images from the CUB dataset, which indicate fewer parts. We also extract the features after the last bottleneck of the ResNet50 encoder similar to our method to test the score of classifcation using logistic regression. We get a score of $\approx 10\%$. Note that we initialize with ImageNet DetCon model before training which has a score of $\approx 35\%$. The performance deteriorates while training PiCIE since it is not able to extract parts using last layer features and global clustering across the dataset. More importantly, PiCIE assigns global cluster centres for the whole dataset. This means that birds are not clustered independently, so different birds get treated the same and segmented into the same classes.

D. Fine-tuning Results

In Tab. 5 we report the fine-tuning results obtained based on the description and hyperparameters listed in the main paper. Here we report the numbers for models initialized with ImageNet DetCon weights before training for part contrast. We report

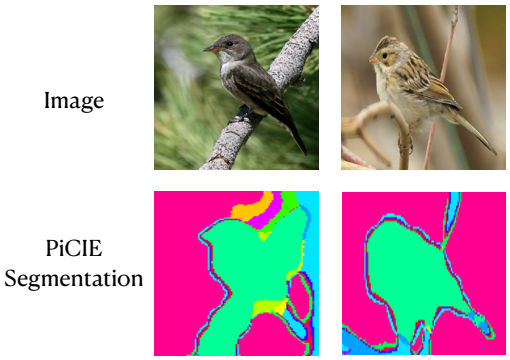

Figure 6. **Visualization of segmentation produced by PiCIE.** We show the segmentation maps produced by PiCIE when trained on the the CUB dataset. PiCIE is mainly able to capture the figure-ground segemntation.

the accuracy over the respective test sets of CUB and FGVC Aircraft.

Method	CUB ft.	FGVC ft.	CUB reg.	FGVC reg.
Imagenet Supervised	76.8	82.8	66.3	46.5
ImageNet DetCon (self-supervised)	62.0	78.7	35.4	35.5
PARTICLE w/ Color + Texture	65.6	79.3	37.1	40.7
PARTICLE w/ Hypercol. (DetCon)	68.9	**81.0**	40.9	**43.9**
PARTICLE w/ Side Information	**70.0**	78.7	**43.7**	49.0

Table 5. **Performance on Fine-tuning using ResNet50.** We show the accuracy on fine-tuning (ft.) on fine-grained classification and compare to logistic regression accuracies (reg.). For Aircrafts the accuracy in fine-tuning is much closer to ImageNet supervised models.

E. Effect of Number of Clusters

We vary the number of clusters for Step I - the part discovery and visualize the clusters produced in Fig 7. We train using the CUB dataset following the same hyperparameters as described in the paper for all variations of number of clusters. We report the scores on the downstream tasks of classification and part segmentation on CUB dataset in Tab 6.

	k=10	k=15	k=20	k=25	k=30
Classification	36.73	38.00	40.11	**40.88**	40.64
Segmentation	47.35	48.49	49.02	49.23	**49.34**

Table 6. **Effect of number of clusters.** We vary the number of clusters on hypercolumns of DetCon ImageNet and test the performance on classification on the CUB dataset. There is a sharp increase when k changes from 10 to 20, after which performance is relatively stable.

F. Part-wise Segmentation of ResNet vs ViT

We show the mean IoU of each part for CUB few-shot segmentation task on the test set for both ImageNet pre-trained DetCon and ImageNet pre-trained DINO. DINO performs much better in finer parts such as eyes and legs. Also the background foreground segmentation is better for DINO based model.

G. Self-Attention of ViTs using DINO and DeiT

DINO's self-attention maps has greater support on the foreground object compared to supervised ViTs trained using DeiT as seen in Fig. 8. Even though both DINO and DeiT has been trained on ImageNet which has very few aircraft images, DINO

Model	bg	head	beak	tail	left wing	right wing	left leg	right leg	left eye	right eye	body	mean
DetCon ResNet	95.38	63.10	43.72	49.50	43.65	54.81	19.75	25.78	27.40	39.24	59.24	47.42
DINO ViT S/8	96.07	64.54	48.20	55.64	48.89	47.41	13.11	25.99	37.66	46.48	61.36	49.58

Table 7. **Part-wise segmentation performance of DetCon ResNet vs DINO ViT.** DINO ViT performs much better in smaller parts such as eye or legs, possibly because it does not spatially downsample features.

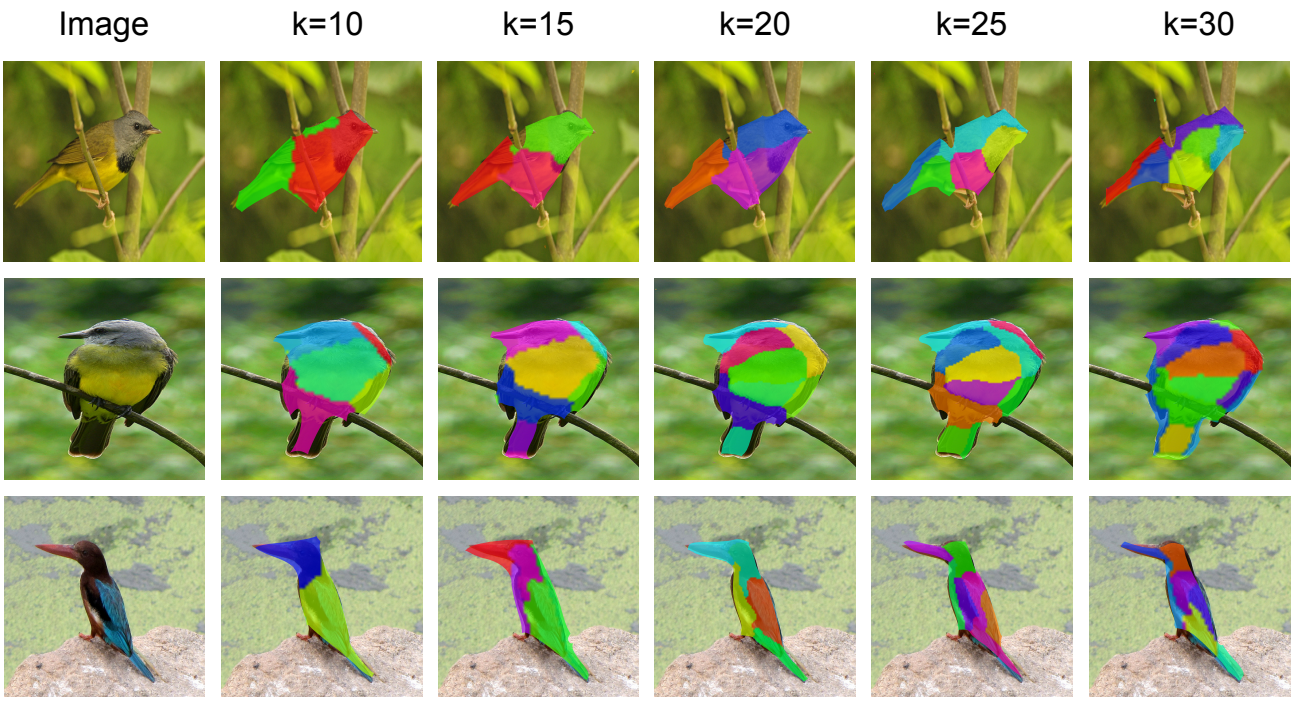

Figure 7. **Visualization of DetCon ImageNet features clusters varying the number of clusters in k-means.** Only the foreground clusters are shown by masking the background clusters as stated in Fig 2. We choose k=25 for experiments reported in the main paper.

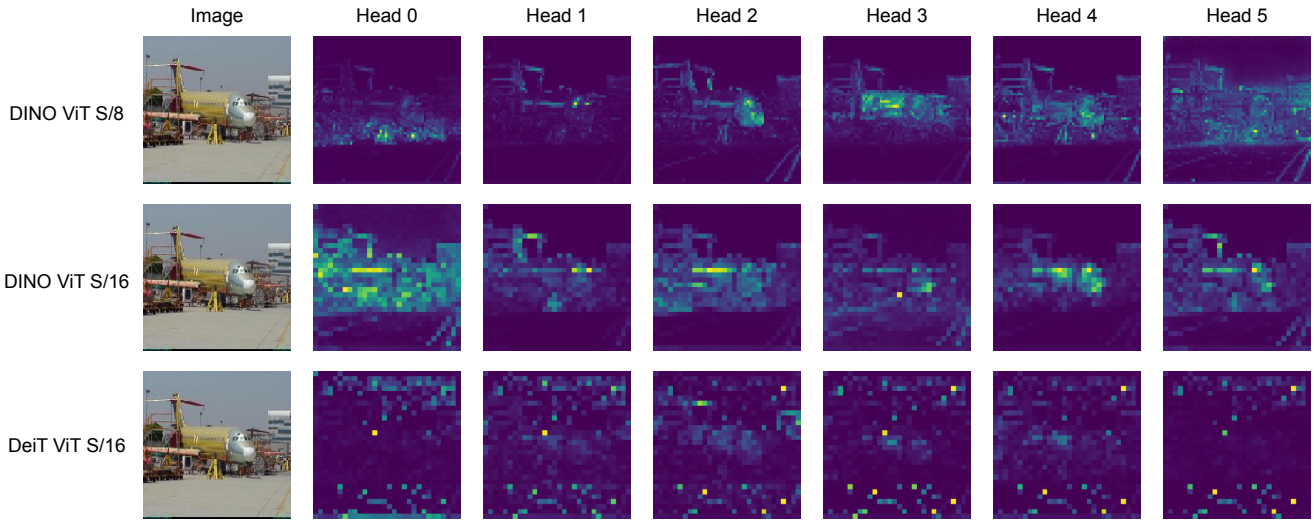

Figure 8. **Comparison of self-attention for ViT trained with DINO vs DeiT.** We visualize the attention maps of the `[cls]` token for the 6 heads of DINO variants and supervised model DeiT [36]. DINO models better localize the aircraft parts than the DeiT models.

still is able to localize the foreground object unlike DeiT. We compare small versions of each model. For DINO the 8×8 model finds finder details than the 16×16 model.

H. Effect of DINO ViT patch size

In Tab 8 we show the performance of DINO ViT S/16 baseline (pre-trained on ImageNet) and when fine-tuned on CUB/OID using DINO objective vs PARTICLE. We compare it with the numbers over DINO ViT S/8 we reported in Tab 1. ViT S/16 baseline perform worse than ViT S/8 in classification and considerably in segmentation because of its coarser patch

size. Again in the case of DINO ViT S/16, PARTICLE offers favorable gains over the baseline.

Method	Arch.	CalTech-UCSD Birds		FGVC Aircrafts	OID Aircrafts
		Cls	Seg	Cls	Seg
DINO	ViT S/8	83.36	48.60 ± 1.48	72.37	60.75 ± 0.35
DINO ft		83.36	48.73 ± 0.61	72.37	60.73 ± 0.48
PARTICLE ft		84.15	50.59 ± 0.79	73.59	61.68 ± 0.59
DINO	ViT S/16	81.25	43.78 ± 0.83	63.12	54.34 ± 0.85
DINO ft		81.82	44.09 ± 0.96	63.94	55.51 ± 1.06
PARTICLE ft		83.02	46.25 ± 0.56	65.43	57.80 ± 1.14

Table 8. **Comparison of patch size of DINO ViT.** We compare DINO ViT S/8 which uses 8×8 patches with DINO ViT S/16 which uses 16× patches. DINO ViT S/16 performs worse than S/8 particularly in the segmentation task.