Location-Aware Self-Supervised Transformers for Semantic Segmentation

Mathilde Caron Neil Houlsby Cordelia Schmid
Google Research

Abstract

*Pixel-level labels are particularly expensive to acquire. Hence, pretraining is a critical step to improve models on a task like semantic segmentation. However, prominent algorithms for pretraining neural networks use image-level objectives, e.g. image classification, image-text alignment à la CLIP, or self-supervised contrastive learning. These objectives do not model spatial information, which might be sub-optimal when finetuning on downstream tasks with spatial reasoning. In this work, we pretrain networks with a **loc**ation-**a**ware (LOCA) self-supervised method which fosters the emergence of strong dense features. Specifically, we use both a patch-level clustering scheme to mine dense pseudo-labels and a relative location prediction task to encourage learning about object parts and their spatial arrangement. Our experiments show that LOCA pretraining leads to representations that transfer competitively to challenging and diverse semantic segmentation datasets.*

1. Introduction

The spatial annotations required for training semantic segmentation models are extremely time consuming and costly to acquire [72]. Therefore, pretraining is commonly used to improve performance and label-efficiency of these models [54]. The dominant method for pretraining neural networks uses image-level tasks on massive amounts of supervised data [11, 46, 49, 67, 70]. For example, powerful foundation models such as Flamingo [1], CoCa [66] or PaLI [13], build upon a visual encoder pretrained by matching aligned image and text pairs with a contrastive loss [46], or by classifying images into a predefined set of categories [70]. These two standard supervised pretraining objectives operate at the global (whole image) level, without explicitly encouraging spatial reasoning.

However, it is unclear whether image-level pretraining is the optimal strategy when targeting recognition tasks with spatial understanding such as semantic segmentation. In fact, a recent study by Minderer *et al*. [40] shows that some models pretrained with image classification, while being excellent at image-level downstream tasks, transfer poorly to

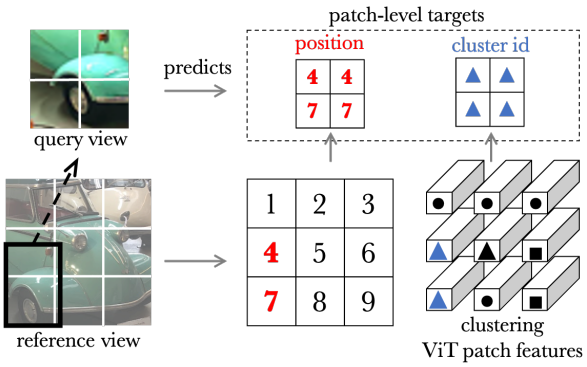


Figure 1. **LOCA** is a self-supervised pretraining method which combines relative position and patch-level cluster prediction. This achieves improved transfer on semantic segmentation datasets.

object detection, a task also requiring spatial reasoning. We argue that the main reason why pretraining is usually done with global objectives is because annotations are much easier to collect at the image level rather than at the pixel level. Indeed, the image classification or image-text datasets typically used in state-of-the-art systems [13, 46, 51, 70] are orders of magnitude bigger and cover more categories than densely annotated datasets [24, 34, 36, 72]. For example, while being much larger than previous densely annotated datasets, the recent Segment Anything dataset [34] remains relatively small (11M images with category-agnostic segmentation masks) compared to standard datasets used for visual pretraining like LAION-{400M, 5B} [51] or WebLI [13]. Therefore, one approach to unlock the potential of dense, spatially-aware pretraining at scale might be to move away from annotations altogether, as proposed by self-supervised learning (SSL) approaches. A successful branch of SSL, often coined as "contrastive learning", works by matching the representation of different views obtained from a same image by means of data augmentation [9, 12, 27, 30]. Interestingly, Caron *et al*. [10] have shown that segmentation masks emerge from the attention maps of Vision Transformers (ViT) [22] trained with these contrastive methods and several works have built on this observation to generate completely unsupervised segmentations [28, 52, 74]. However, we found in our preliminary experiments that salient attention maps do not correlate with superior performance after *finetuning* to the se-

mantic segmentation task [74]. We hypothesize that this is because contrastive methods operate at the global level without explicitly encouraging spatial relationships. Worse, some works [3, 15] have analyzed that due to the intensive use of spatial data augmentation (cropping, rescaling, etc) of these methods, they tend to produce *localization-invariant* features, discarding spatial information.

Hence, in order to foster the emergence of strong dense representations, our goal is to design a patch-level pretext task encouraging spatial localization reasoning. Recently, patch-level SSL pretrainings have attracted more and more attention in the community [4, 5, 20, 29, 60, 64, 69]. For example, dense contrastive approaches adapt the popular contrastive SSL paradigm to the patch level [45, 48, 58, 62, 63] while masked autoencoders propose to reconstruct masked patches [5, 29]. Of particular interest, Zhai *et al*. [69] propose a pure localization method, that of predicting the position of the patches of an image. Intuitively, position prediction should inherently require a strong spatial and semantic understanding and has been the core motivation of the pioneering SSL branch of "jigsaw puzzle" [21, 42]. In this work, we propose to revisit this strategy and introduce a relative position prediction task. Specifically, our method works by predicting the location of a *query* view relatively to another, *reference*, view. To be able to locate themselves in the reference, the query patch features "look" at those of the reference through shallow cross-attention. We control the difficulty of the task and properties of the resulting features by masking reference patch features visible to the query. Our experiments show that this query-reference mechanism improves greatly over the single-view design of Zhai *et al*. [69] when transferring to semantic segmentation.

Since semantic segmentation is a per-patch classification problem, we also propose to prepare the ViT features for this task by means of clustering-based pseudo-labeling [2, 8, 65] done at the patch level [74]. Overall, we present in this work a **loc**ation-**a**ware (LOCA) self-supervised pretraining approach for semantic segmentation, which combines a straightforward patch-level SSL clustering method and relative position pretraining, as illustrated in Fig. 1. We show that LOCA yields improved performance over state-of-the-art supervised [46, 53, 56] and unsupervised [10, 14, 29, 73] representation learning methods for ViTs when transferred to 11 diverse semantic segmentation benchmarks. We summarize our main contributions: (i) novel location-aware SSL methodology; (ii) performance lifts in several downstream semantic segmentation (and depth estimation) tasks; (iii) promising scaling capabilities in model and data axes; (iv) comprehensive ablations studies of our relative position prediction framework.

2. Related Work

SSL with location prediction. Pioneering works in SSL proposed to exploit spatial cues to generate pretext tasks [21, 26, 33, 35, 41, 42, 50]. Notably, inspired by word2vec [39], Doersch *et al*. [21] train a network to predict the relative position of a pair of patches from the same image while Noroozi and Favaro [42] extend this approach to solving "jigsaw puzzles" by rearranging a set of shuffled crops of an image. These approaches were developed with Convnets and only very little work [69] has revisited them in the scope of Transformers. Zhai *et al*. [69] propose to pretrain a ViT to predict the position of its input patches given their visual appearance only, i.e. by discarding positional embeddings. We compare this strategy to the LOCA relative position mechanism in Fig. 2 and Sec. 4.1. Also using localization, UP-DETR [18] propose to pretrain the entire object detection DETR architecture [7] (with a transformer encoder-decoder, backbone, object queries, etc...) by localizing random boxes in a reference image. Two key methodological differences in the *position* prediction tasks of UP-DETR and LOCA are: we use (i) masking to increase the difficulty of the task and (ii) a different localization loss. We evaluate the impact of these two components in Sec. 4.1.

Context and masked auto-encoders. Also exploiting spatial cues for SSL, Pathak *et al*. [44] propose context auto-encoders to train Convnets to generate the content of a masked region based on its surrounding. Masked auto-encoders have revisited this "inpainting" approach to pretraining ViTs [5, 29, 59]. Specifically, the task is to reconstruct masked [5] or dropped [29] patches from the input sequence tokens, either directly in pixel space [29] or in feature space [5, 59, 73]. Similar to LOCA, masked auto-encoders are trained with patch-based objectives with a task encouraging learning local representations. We compare the performance of these two paradigms in semantic segmentation transfer in Sec. 5.

Clustering-based SSL uses clustering while training to mine pseudo-labels in a dataset without annotations [8, 65]. This pseudo-assignment strategy is usually done at the image level [2, 3, 8, 9, 65] but recent works such as Leopart [74] propose to cluster patch-level representations instead to produce dense semantic features [16]. The clustering pipeline of LOCA is simplified compared to Leopart since we empirically find that design choices such as ROI align or foreground focusing are not useful in our setup while complexifying the implementation. Another difference of LOCA with Leopart is that we find that we can train from scratch without the need for initialization from an external pretrained checkpoint [10]. Finally, unlike Leopart, our work leverages an explicit position-based pretraining. Quantitative comparisons with Leopart are in Sec. 5 (see Tab. 3).

Dense contrastive SSL. A prominent line of SSL, often referred to as "contrastive" or "siamese" approaches, trains

networks by matching the representation of different views obtained from a same image by means of data augmentation [3, 9, 12, 23, 27, 30, 61]. These approaches have primarily been developed with global (image-level) objectives but several recent works have adapted them to learn local features [20, 32, 45, 48, 58, 62, 63, 68]. Specifically, instead of matching representations from global descriptors, they match features that come from the same location in the original image but seen from different views [45]. We borrow the strategy of tracking the data augmentation process of two views to find their intersection [45]. LOCA differs from this body of work by exploring a different task, that of predicting relative positions. Another difference is the use of a patch-level loss based on *clustering* rather than *contrastive* self-supervision, which recent work Leopart [74] has shown to be more effective.

Unsupervised semantic segmentation. While our goal is to improve pretraining for semantic segmentation, some parallel works to ours directly target semantic segmentation without using any supervision at all [16, 57]. Indeed, [10] have shown that unsupervised segmentation masks emerge from the attention module of ViTs trained with image-level contrastive objectives such as DINO. Several works build on this observation and enhance SSL features to produce completely unsupervised segmentations [28, 52, 74].

3. Methodology

In order to foster the emergence of strong dense representations for semantic segmentation downstream tasks, we design LOCA, a patch-level SSL pretraining task that requires to reason about spatial localization. It works as a query-reference scheme where patches of a query view predict both their position and their cluster assignment relatively to a reference view, as illustrated in Fig 1.

Generating query and reference views. From an image x of a dataset, we form a *reference* view (denoted by x_{ref}) and a *query* view (denoted by x_q) using a randomized data augmentation routine composed of flipping, cropping, rescaling and color jittering. Because query and reference are generated by two independent augmentation draws, they usually have different image statistics (i.e. different scale, region or color histogram). This forces the network to rely less on low-level cues (chromatic aberration, color, and edge consistency) to solve the self-supervised task and more on recognizing object parts and their organization.

The query's predictions are supervised by the reference view and therefore our loss is defined only at the intersection of the two views. Hence, we want the query and reference to intersect often. Also, we wish to constrain the spatial extent of the queries in order to favor the emergence of image-*part* representations. A natural choice then is to sample the reference view so that it covers a large area of the original image and the query views so that they cover a small portion of the original image. In practice we use random resize-cropping and patch-dropping (following the input pipeline of MSN [3]) for generating different query views per reference. We describe the method for a single query for simplicity but use ten in our experiments.

Correspondences between query and reference. Following the standard protocol of ViTs [22], query and reference views are divided into non overlapping patches of resolution $P \times P$. More precisely, the reference view is flattened into $N_{ref} = \lfloor H_{ref}/P \rfloor \times \lfloor W_{ref}/P \rfloor$ (with $H_{ref} \times W_{ref}$ the resolution of x_{ref}) separate patches $x^i_{ref}, i \in \{1, ..., N_{ref}\}$. Default values are $H_{ref} = W_{ref} = 224$, $P = 16$ and $N_{ref} = 196$. A similar "patchification" process is applied on the query view, resulting in a sequence of N_q patches $x^j_q, j \in \{1, ..., N_q\}$, where we typically use $N_q = 36$. By tracking the data augmentation draws that generated x_{ref} and x_q, we can identify the patch-level correspondences between these two views. In particular, we know a function h that, given any patch position j in the query view, returns the position $i = h(j)$ of the patch in the reference, $x^{h(j)}_{ref}$, that has the greatest overlap with the query patch x^j_q. We implement the function h with successive nearest interpolations and because the patchification grids of x_q and x_{ref} are usually not exactly aligned, a pair of matching patches, x^j_q and $x^{h(j)}_{ref}$, have similar content but do not generally match *perfectly*. For example, we can see in Fig. 1 that the bottom left patches of both the query and references are in correspondence but do not cover exactly the same content.

Patch-level encoding with ViT. We process both the reference and query views with a ViT [22] denoted by f of internal dimension d ($d = 768$ for ViT-B). We note $Z_q \in \mathbb{R}^{d \times N_q}$ (resp. $Z_{ref} \in \mathbb{R}^{d \times N_{ref}}$), the output patch-level representation matrix of the query (resp. reference) view.

Patch position prediction. To encourage the network to learn about different object parts and their spatial arrangement, we predict relative patch positions. We implement a query localization problem as a N_{ref}-way classification task where each query patch representation has to predict the position of its corresponding patch in the reference view, as given by h. To that end, the patch representation of the query needs to "look" at those of the reference. We implement this query-reference interaction with a single cross-attention transformer block, denoted by g, whose queries are computed from Z_q and keys and values are obtained from Z_{ref}. We denote the query representations after they have looked at the reference as $G = g(Z_q, Z_{ref}) \in \mathbb{R}^{d \times N_q}$ and by $W \in \mathbb{R}^{d \times N_{ref}}$ the final position classification layer. Note that N_{ref} is the total number of positions in the reference. We minimize the position classification loss:

$$\frac{1}{|\Omega|} \sum_{j \in \Omega} \ell((W^T G)_j, h(j)) \quad (1)$$

where Ω is the set of patch position in the query that has an

intersection with the reference (i.e. where h is defined) and ℓ is the softmax cross-entropy loss.

Masking reference patch features visible to the query. In practice, we find that problem 1 can be solved near perfectly by the network (see the validation accuracy in Fig. 3). As empirically shown in Sec. 4.1, one strategy to make the task more challenging is to restrict what the query can see from the reference. We implement this mechanism by randomly masking a ratio η of the patch features input to the cross-attention block g. Specifically, we redefine $G = g(Z_q, m(Z_{ref}, \eta))$ where m is a random process that drops $\lfloor \eta N_{ref} \rfloor$ columns of Z_{ref}. We use structured dropping (i.e. we keep a consecutive subset of patch tokens) as we find in our experiments that it leads to superior performance than unstructured dropping (+0.8 mIoU).

Patch-level clustering. Training for semantic segmentation in a supervised setting is typically cast as a per-patch classification problem over K predefined categories:

$$\frac{1}{N_q} \sum_{j=1}^{N_q} \ell((Q^T Z_q)_j, y_j)$$

where Q is a matrix in $\mathbb{R}^{\tilde{d} \times K}$ of learnable category prototypes and ℓ is the softmax cross-entropy loss. This problem is supervised by patch-level annotations y_j. However, because we do not have access to such annotations, we resort to clustering for pseudo-supervision [8, 9]. In particular, to supervise the patch j in the query, we cluster the patch representations of the corresponding reference view into K clusters, playing the role of pseudo-categories. We obtain a soft cluster assignment (or pseudo-label) based on the similarity between the prototypes and the patch representation at the corresponding localization in the reference view:

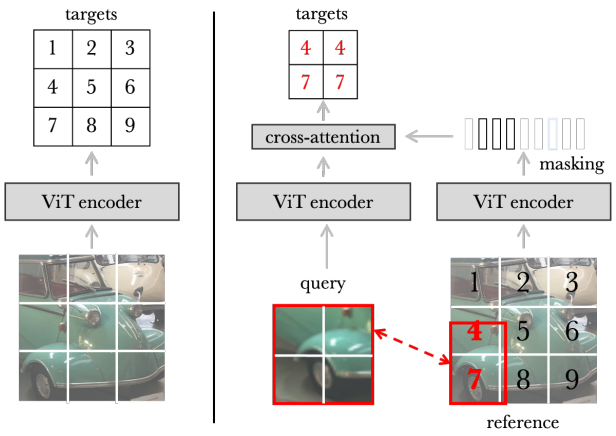
$$y^i_{ref} = \text{softmax}(\tilde{Z}^i_{ref}.Q/\tau)$$

with $i = h(j)$ and τ a temperature parameter controlling the sharpness of the distribution. Note that, as commonly done in SSL [9, 12, 27], we have projected the representations Z_q and Z_{ref} with a 2-layer multilayer perceptron (MLP), resulting in features $\tilde{Z}_q \in \mathbb{R}^{\tilde{d} \times N_q}$ and $\tilde{Z}_{ref} \in \mathbb{R}^{\tilde{d} \times N_{ref}}$ with $\tilde{d} = 256$. We further adjust the cluster assignment distribution with Sinkhorn-Knopp [17] to avoid the collapsing trivial solution [2, 9]. Since we have replaced expensive per-patch label supervision with cluster pseudo-labels we can minimize the following objective:

$$\frac{1}{|\Omega|} \sum_{j \in \Omega} \ell((Q^T \tilde{Z}_q)_j, y^{h(j)}_{ref}) \quad (2)$$

where Ω is defined as in Eq. 1. We regularize this loss function with the mean entropy maximization (me-max) protocol [3] to encourage the network to use the full set of pseudo-label prototypes Q (see Tab. 2a)).

Optimization. We train LOCA by minimizing the sum of the objectives in Eq 1 and Eq 2, with equal weighting and averaged both over the different query views and the minibatch. We learn the parameters of f, g, Q, and W by back-propagating in the branch processing the query views. The parameters used in the branch processing the reference views are updated via an exponential moving average of the encoder parameters processing the query views [10, 27, 30]. We find that this asymmetry does not have any effect on the position prediction but improves performance and stability for the cluster prediction task.



(a) Single (b) LOCA query-reference (Ours)

Figure 2. **Conceptual comparison of single vs query-reference** patch position prediction mechanisms: (a) in a single view as in Zhai *et al*. [69]; (b) in a query view relatively to a reference view as in LOCA (Ours). Quantitative comparison is in Fig. 3. Masking not illustrated for single.

Implementation and evaluation. We train LOCA on ImageNet datasets without labels with learning rate of 0.001 (cosine schedule), batch size of 1024 and weight decay of 0.1 with adamw [37]. Models in Sec. 5 are trained for 600 epochs and those for analyses (Sec. 4) for 100 epochs. We evaluate by end-to-end finetuning on 11 semantic segmentation benchmarks [38], detailed in the Appendix. We follow and reproduce the linear decoder protocol of [54]. It uses a minimal amount of adapter layers to prevent the effect of pretraining of being washed out by heavy decoders [43]. We report results for other methods if available and run evaluation from publicly released checkpoints if not. We run a hyperparameter search with the same budget for all methods. We report results in single scale, averaged over 5 runs. All implementation details are in the Appendix and code will be released.

4. Design Choices Analyses

In this section, we detail various design choices for LOCA. First, we make an in-depth study of the position prediction. Second, we present an ablation study focused on the pseudo-labeling clustering technique.

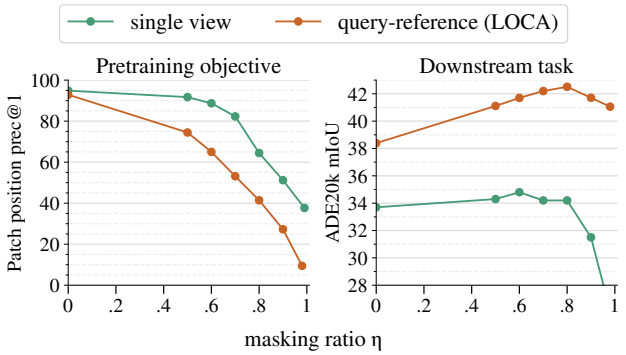


Figure 3. **Single vs query-reference** patch position prediction mechanisms. For both mechanisms, we report the position prediction accuracy (left) and the performance after transfer to semantic segmentation on ADE20k (right) for different patch masking ratios η. Query-reference makes for a more challenging pre-training objective (lower accuracy on the position prediction task) due to different image statistics between query and reference and constrained patch interactions. Conceptual differences are illustrated in Fig. 2. Varying the masking ratio controls the difficulty of the task and improves transfer performance.

4.1. Position prediction framework

To encourage the network to learn about the spatial arrangement of different object parts, we propose to predict relative positions. We detail here different components of our framework: the query-reference mechanism, the effect of masking reference patches and the loss function. Unless specified otherwise, models are trained solely with loss (1) in this section to isolate the effect of position-based training.

Query-reference. We compare the two mechanisms illustrated in Fig. 2. The "single" strategy is akin to Zhai *et al*. [69]. We vary η the proportion of masked patch tokens. In "single", masking patch tokens means that patches can only attend to the unmasked ones, i.e. only the unmasked patches take part in the computation of attention keys and values [69]. Results are in Fig. 3. On the left, we report the validation accuracy for the position prediction task, which measures the difficulty of this task. On the right, we show transfer performance.

We see in Fig. 3 that the query-reference mechanism of LOCA is a more challenging pretraining framework than Zhai *et al*. [69] and subsequently leads to better representations for semantic segmentation (+7.6 mIoU). This can intuitively be explained by several conceptual differences. First, in [69], the network can almost perfectly solve the task by leveraging low-level non-semantic cues such as chromatic aberration, color or edges consistency between patches. This is *partly* prevented in the query-reference mechanism due to different image statistics between query and reference, thanks to cropping, rescaling and color jittering. We evaluate the effect of color jittering alone in the Appendix. Second, the way query (i.e. patches that predict a position) and reference (i.e. context patches) can interact is in stark contrast in the two mechanisms. In [69], query and reference interact in an unconstrained manner at all stages of the computation since they are the same entity. With masking, this design is party modified by processing each query patch independently but still allowing them to fully attend to the reference patches at each block. By contrast, in LOCA, query patches can attend freely to each other but cannot look at the reference patches until the last stage of the network. Intuitively, this more constrained interaction encourages both query and reference patches to develop stronger final localization features.

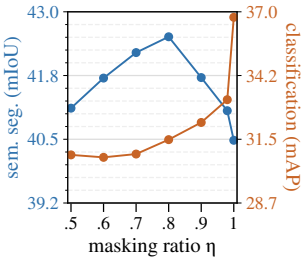


Figure 4. **Masking reference patches to the query** improves both classification and segmentation on ADE20k. Too much masking prevents the query from solving the task by spatial reasoning which hurts segmentation.

Masking reference patches. In Fig. 3, we observe that the localization pretraining task can be solved near perfectly when all the patches in the reference are visible to the query (see validation accuracy in Fig. 3 left for $\eta = 0$ and first column of Fig. 5 for visual examples). Masking patches to the query makes the pretraining objective more challenging and leads to better representations. In Fig. 4, we analyze this effect further. We consider different masking ratios and report for the same downstream dataset both the transfer performance on semantic segmentation and multi-label classification (with frozen backbone) by turning the semantic segmentation annotations into classification labels. We observe in Fig. 4 that masking improves both localization and classification. Intuitively this is because masking reference patches forces the query to rely less on finding matching salient points between the two views and more on recognizing objects and their parts as illustrated in Fig 5.

However, when masking is too aggressive, we observe that the query does not see enough of the reference to solve its task by relative localization and resorts to other cues. To understand this phenomenon, we push masking to extreme rates and even report performance when the reference is *not visible at all* ($\eta = 1$). Surprisingly, we find that the query still manages to solve the localization pretraining task to some extent with a localization accuracy of 3.7% (random guessing achieves 0.5%). We hypothesize that two ways of solving the task without looking at the reference are to (i) learn where things are typically located in images and (ii) memorize all the dataset images. We argue that the "memorization" regime is akin to an implicit formulation of the "exemplar" instance discrimination approach of Dosovitskiy *et al*. [23] where the network learns to recognize each individual instance of a dataset (but without a classifier of the size of the dataset as in [23]). Overall, both learning bi-

Output	Predicts spatial extent	Loss	ADE20k
All patch positions	✓	Classif.	42.5
Central patch position		Classif.	38.6
Box coord. (UP-DETR [18])	✓	Regress.	39.0

Table 1. **Localization loss.** We report mIoU on ADE20k for different loss variants. Predicting the position of all patches *vs* the position of the central patch only is better, likely because it involves reasoning about the spatial extent of the query.

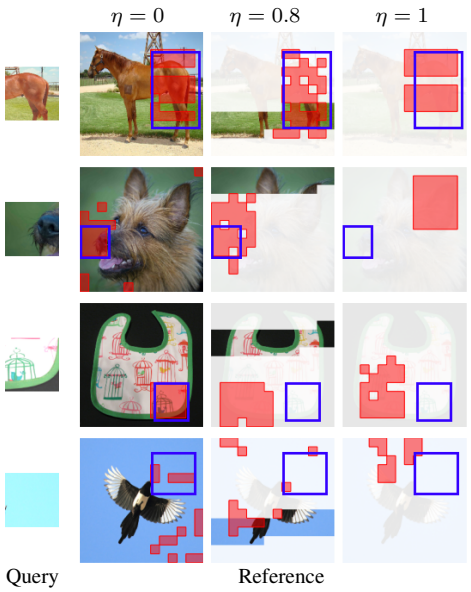


Figure 5. **Visualizing LOCA's predictions.** The query location is shown in blue in the reference and LOCA predictions are shown in red. Columns correspond to different reference masking rates and we show only patches visible to the query. More examples in Appendix. Displayed images are not seen during training.

ases of general dataset statistics and instance discrimination have been shown to improve transfer performance on classification downstream tasks [23, 25, 61] which is consistent with the boost in classification observed for $\eta = 1$.

Overall, this experiment shows that an optimal masking ratio for semantic segmentation features is high, but not too high either so that the network can still solve the task by *relative localization*. In practice, we use $\eta = 0.8$.

Choice of localization loss. First, we compare predicting the position of all patches versus the position of the central patch only. We see in Tab. 1 that all patches is better. We hypothesize that this is because it requires to predict the spatial extent of the query and not just an anchor point. Second, we compare solving a per-patch position classification problem versus regressing the coordinates of the query box in the reference. For box prediction, we use a linear combination of ℓ_1 loss and the generalized IoU loss, following UP-DETR [7, 18]. Because query and reference patchification grids are usually not aligned, matching patches in query and reference do not have exactly the same content. This does not affect the box regression formulation, which might give it an advantage over per-patch classification. However, we surprisingly find in Tab. 1 that box regression leads to poorer performance than per-patch classification. Our hypothesis is that the jittering induced by the grid misalignment regularizes the training while exact box regression encourages to focus on precise but low-level cues.

a) SK and me-max encourage use of all prototypes. H: average prediction entropy.

SK	me-max	H	ADE20k
✓	✓	8.28	46.2
✓		8.27	46.1
	✓	8.14	45.7
		0	collapse

b) Patch-level clustering is better for transfer on semantic segmentation.

Cluster	classif. Im1k-10s	semseg. ADE20k
Patch	40.9	46.2
Image (CLS)	48.6	43.8
Image (GAP)	48.5	43.8

c) Effect of the number of cluster prototypes.

K	2^4	2^8	2^{12}	2^{14}
ADE20k	45.3	46.2	46.2	45.8

d) Number of queries.

# queries	10	5	2	1
ADE20k	46.2	45.5	43.4	41.1
Speedup	–	×1.5	×2.1	×3.0

Table 2. **Ablation study of different design choices.**

Visualizing LOCA's predictions. In Fig. 5, we visualize the output of location prediction models trained with different masking rates: $\eta = 0$ (no masking), $\eta = 0.8$ (default) and $\eta = 1$ (invisible reference). The first row shows a situation where the network can make a valid guess about the query's location solely based on the query visual appearance, i.e. without looking at the reference. In the second row, we see that LOCA successfully locates the snout of the dog based on the reference ear patches. This suggests that it has learned about spatial arrangement of different parts of a dog. Third row depicts the case where the network can leverage low-level cues such as edge consistency to locate the query. The masked variants are restrained in their use of such cues and hence fail to locate the query. Finally, in last row, there is no visible cue in the query that allows its localization. The prediction is degenerated for all the variants.

Combining with patch clustering. In the previous experiments, we have validated our position prediction scheme and showed that it improves by +7.6mIoU over the position prediction method of Zhai *et al.* [69]. While we find that predicting position only is performing less well than predicting patch-level cluster assignments only (−3.3mIoU) the best performance is obtained when predicting *both* (+0.7mIoU over cluster only) which demonstrates some complementary between them.

4.2. Ablation study of patch clustering

In this section, we report model ablation results focused on the clustering mechanism. In Tab. 2 a), we see that both Sinkhorn-Knopp and me-max regularizations are useful to

Method	dense?	*Consumer* ADE20k	P.Cont	P.VOC	*Driving* Citys.	BDD	CamVid	IDD	KITTI	*Indoor* SUN	*Aerial* ISPRS	*Underwater* SUIM	Avg. rel. Δ (%)
ImageNet-1k - ViT-Small/16													
Random init.		20.4	20.7	32.1	43.7	39.2	41.6	44.0	38.2	18.9	35.6	56.1	0
DINO [10]		41.2	46.7	72.7	69.3	58.9	51.7	52.8	45.0	42.4	40.3	70.7	62.6
MoCo-v3 [14]		42.5	49.3	72.0	69.0	59.0	51.8	53.3	45.2	44.2	40.4	**73.6**	65.6
Leopart [74]	✓	42.2	48.7	73.3	70.8	59.1	52.0	53.3	46.1	43.4	40.7	71.1	65.5
iBOT [73]	✓	42.6	49.7	**74.5**	**71.3**	60.1	53.1	54.4	46.7	44.7	**45.8**	72.2	69.5
LOCA (Ours)	✓	**44.8**	**51.3**	74.0	70.9	**60.5**	**56.6**	**55.0**	**47.9**	**45.2**	43.0	73.5	**72.0**
ImageNet-1k - ViT-Base/16													
Random init.		21.1	19.6	29.1	51.4	40.2	43.3	45.2	39.0	19.7	28.1	53.0	0
DINO [10]		44.1	50.7	74.1	78.4	60.7	51.5	54.3	46.4	44.4	41.5	71.2	71.9
MoCo-v3 [14]		45.4	51.6	74.5	78.6	60.4	51.1	53.7	45.7	45.6	42.1	72.6	73.6
iBOT [73]	✓	47.0	54.6	75.0	**79.8**	62.1	51.5	55.5	47.0	46.3	42.2	73.2	77.7
MAE [29]	✓	45.5	51.7	75.0	79.7	62.1	**57.8**	**55.8**	48.3	45.9	44.6	72.4	77.8
LOCA (Ours)	✓	**47.9**	**54.9**	**76.7**	**79.8**	**62.8**	56.1	55.6	**48.5**	**47.7**	**45.6**	**74.0**	**82.1**

Table 3. **Comparison with other SSL pretrainings on 11 semantic segmentation benchmarks.** We report mean IoU on the different validation sets. In the last column, we report the relative improvement over starting from random initialization averaged across all the datasets. We consider SSL methods trained on ImageNet-1k without labels using the ViT-Small and -Base architectures.

encourage the model to use the full set of cluster prototypes. In Tab. 2 b), we compare our patch-level pseudo-labeling method to image-level ones on ADE20k semantic segmentation and on ImageNet-1k 10-shot classification. The image-level clustering framework is akin to existing SSL frameworks such as DINO [10] or MSN [3]. We evaluate two global aggregation techniques: token (CLS) and global average pooling (GAP). We see that performance on semantic segmentation improves with per-patch assignments instead of image-level clustering. However, we observe a decay on classification. In Tab. 2 c), we see that LOCA is robust to the number of clusters K, though over-clustering is beneficial [8, 74]. In Tab. 2 d), we show the effect of reducing the number of queries. Using a single query instead of 10 allows to speed up pretraining time by $\times 3$ but induces a loss of 5.1mIoU in transfer performance.

5. Main Results

5.1. Comparison with other SSL pretrainings

In this section, we compare LOCA to popular state-of-the-art SSL models for ViTs: DINO [10], Leopart [74], MoCo-v3 [14], MAE [29] and iBOT [73]. Compared models use ImageNet-1k (without labels) and ViT-{B, S}/16.

Transfer to 11 semantic segmentation benchmarks. In Tab. 3, we report the performance of different SSL pretraining methods after end-to-end semantic segmentation finetuning on diverse datasets. First, we see that representations learned with LOCA transfer very well to semantic segmentation across the different considered datasets and architectures. Of particular interest, MAE achieves the second best SSL performance. In terms of training efficiency, one LOCA epoch takes 17.4 minutes while one MAE epoch takes 5.7 minutes based on our implementation. LOCA reaches a relative improvement of 82.1% in 600 epochs while MAE reaches 77.8% in 2.6× more epochs (1600). Hence, LOCA improves over MAE by +4.3 points while being 1.1× longer to train. We also include in the Appendix a preliminary comparison with recent and concurrent DINO-v2 [43]. Note that DINO-v2 combines image-level losses with a patch-level objective akin to the iBOT objective [73]. Hence, given that LOCA outperforms iBOT across the different tasks in Tab. 3 (+2.5 for ViT-S and +4.4 for ViT-B), a promising direction could be to use LOCA instead as a patch-level objective within DINO-v2 framework.

Method	1/32	1/16	1/8	1/4	1/2	1
Supervised - DeiT-III [56]	20.9	27.1	32.7	38.3	42.0	47.3
DINO [10]	18.4	24.5	29.5	35.2	39.5	44.1
MoCo-v3 [14]	17.7	25.2	30.8	36.5	40.7	45.4
iBOT [73]	20.9	28.0	33.4	38.7	42.6	47.0
MAE [29]	18.4	25.3	30.5	36.1	40.6	45.5
LOCA (Ours)	**22.2**	**30.0**	**34.4**	**39.1**	**42.8**	**47.9**

Table 4. **Fewshot semantic segmentation.** We report mean IoU on the validation set of ADE20k for different SSL pretrained models. All methods use ImageNet-1k and ViT-B/16. Only a fraction of training images are used for finetuning.

Label-efficient semantic segmentation. A good property for pretrained representations is the ability to transfer with few annotations [1, 3, 71]. In Tab. 4 we evaluate features when finetuning on fewshot semantic segmentation. We randomly sample a fraction of training images from ADE20k and use only those to finetune our model [31]. In the 1/32 split, as few as 630 training images are used. We report the average over 5 different folds [31]. We observe that LOCA pretraining improves label-efficiency of seman-

Method	Data	Sup.	dense?	Classif.	Loc.	Both
ViT-Base/16						
CLIP [46]	WIT	Text		58.3	66.4	45.9
AugReg [53]	Im21k	Labels		**60.7**	67.4	48.1
LOCA (Ours)	Im21k	∅	✓	50.2	**68.5**	**48.5**
ViT-Large/16						
AugReg [53]	Im21k	Labels		**60.3**	68.0	50.7
LOCA (Ours)	Im21k	∅	✓	51.6	**71.0**	**52.3**

Table 5. **Comparison with supervised pretrainings** by disentangling localization and classification on semantic segmentation. We report classification only ("Classif.": mAP), localization only ("Loc": mIoU) and full semantic segmentation ("Both": mIoU) on ADE20k. LOCA yields excellent locality understanding.

tic segmentation models. The gap with other methods becomes larger when fewer finetuning images are available.

5.2. Comparison with other pretraining paradigms

In this section, we compare our self-supervised location-aware pretraining to two powerful image-level pretraining paradigms: (i) image classification (i.e. label supervision) as in [53, 70] and (ii) image-text alignment as in CLIP [46].

Localization and classification trade-off. Semantic segmentation combines classification and localization, where these two tasks may have different feature preferences. In Tab 5, we disentangle classification and localization performance for models pretrained with an image- *vs* dense- level objective. For classification evaluation, the task is to predict the label of the masks present in an image but not their spatial extent. For localization evaluation, we use an oracle replacing the label of each mask by the label of the ground truth mask the model has the best IoU with. We report results for ADE20k in Tab. 5 and other datasets in Appendix. We also report results with supervised image classification DeiT-III [56] in Appendix. We observe in Tab 5 that models pretrained with an image-level supervised objective are better at classification than LOCA. However, LOCA is better at pure localization, which results in improved performance on semantic segmentation which requires both locality and class-level understanding.

Depth estimation. The previous experiment (Tab. 5) shows that LOCA features are particularly good at localization. While the focus of this work is semantic segmentation, we explore the potential of LOCA on depth estimation, another per-pixel prediction task requiring high spatial understanding but less semantic understanding. We follow [19] and train a Dense Prediction Transformer [47] with frozen backbone on Waymo Open real-world driving dataset [55]. We observe in Tab. 6 that LOCA transfers better to depth estimation than backbones trained with image-level supervision. LOCA achieves comparable or better performance than supervised ViT-e while having 10× less parameters.

Model	param(M)	MSE ↓	AbsRel ↓	$\delta \uparrow$ < 1.1	< 1.25	$< 1.25^2$
ViT-L sup [53]	304	0.027	0.121	0.594	0.871	0.972
ViT-L LOCA	304	**0.024**	**0.102**	**0.681**	**0.891**	**0.973**
ViT-e sup [70]	3926	**0.024**	0.112	0.631	0.888	**0.975**
ViT-H LOCA	632	**0.024**	**0.101**	**0.685**	**0.894**	**0.975**

Table 6. **Monocular depth estimation** on the Waymo Open dataset [55]. We follow the setup from [19] and report their number for ViT-L and ViT-e supervised ("sup") backbones.

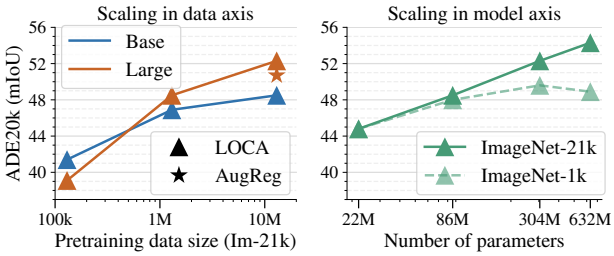


Figure 6. **Scaling study:** transfer on ADE20k validation set.

5.3. Scaling data and model axes

A premise of SSL is that it can scale to arbitrary large datasets since images do not require any annotations. Because location-aware *supervised* pretraining is not feasible in practice due to the huge cost of pixel-level category annotations, we believe our self-supervised spatial pretraining could be a good candidate for scaling. In Fig 6, we propose a scaling study on data and model axes. In the left panel, we see that LOCA Large network benefits more from scaling in dataset size than the smaller Base architecture. In the right panel, we see that pretraining LOCA on full ImageNet-21k scales better in model axis than using smaller, albeit highly curated, ImageNet-1k dataset. Overall, mirroring the trend of image-level supervised pretrainings [13, 70], we observe that we need to scale both dataset size and model capacity to achieve the best of performance. Overall, the results in Fig 6 show that our method scales promisingly to large models and large amount of data, which is a positive signal that it could be a viable candidate for semantic segmentation pretraining at scale.

6. Conclusion

We present a novel self-supervised, spatially-aware pretraining that leads to improved transfer on several semantic segmentation downstream tasks. A promising direction for future work is to combine LOCA dense objective with global image-level ones [6, 43, 73]. Finally, we focused on semantic segmentation (and depth estimation) only in this paper, but the set of visual tasks with localization is large and we hope that our findings can serve as a useful checkpoint for future studies beyond this scope. We discuss potential negative societal impact in the Appendix.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 1, 7

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 2, 4

[3] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 2, 3, 4, 7

[4] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022. 2

[5] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[6] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. In *NeurIPS*, 2021. 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 6

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2, 4, 7

[9] 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. 1, 2, 3, 4

[10] 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. 1, 2, 3, 4, 7

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *preprint arXiv:2002.05709*, 2020. 1, 3, 4

[13] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1, 8

[14] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 2, 7

[15] Yubei Chen, Adrien Bardes, Zengyi Li, and Yann LeCun. Intra-instance vicreg: Bag of self-supervised image patch embedding. *arXiv preprint arXiv:2206.08954*, 2022. 2

[16] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *CVPR*, 2021. 2, 3

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 4

[18] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 2, 6

[19] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023. 8

[20] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Deeply unsupervised patch re-identification for pre-training object detectors. *IEEE TPAMI*. 2, 3

[21] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *preprint arXiv:2010.11929*, 2020. 1, 3

[23] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 2016. 3, 5, 6

[24] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 6

[26] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019. 2

[27] 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, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 3, 4

[28] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022. 1, 3

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2, 7

[30] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 3, 4

[31] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *NeurIPS*, 2021. 7

[32] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Learning where to learn in cross-view self-supervised learning. In *CVPR*, 2022. 3

[33] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *WACV*, 2018. 2

[34] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[35] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 2

[36] 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*, 2014. 1

[37] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 4

[38] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of influence for transfer learning across diverse appearance domains and task types. *IEEE TPAMI*, 2021. 4

[39] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[40] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022. 1

[41] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018. 2

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

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 7, 8

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

[45] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. 2020. 2, 3

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021. 1, 2, 8

[47] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 8

[48] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *CVPR*, 2021. 2, 3

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1

[50] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. In *CVPR*, 2017. 2

[51] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 2022. 1

[52] 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. 1, 3

[53] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 2, 8

[54] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 1, 4

[55] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 8

[56] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 2, 7, 8

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

[58] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2, 3

[59] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022. 2

[60] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In *NeurIPS*, 2022. 2

[61] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3, 6

[62] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *ICCV*, 2021. 2, 3

[63] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021. 2, 3

[64] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 2

[65] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016. 2

[66] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 1

[67] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[68] Sukmin Yun, Hankook Lee, Jaehyung Kim, and Jinwoo Shin. Patch-level representation learning for self-supervised vision transformers. In *CVPR*, 2022. 3

[69] Shuangfei Zhai, Navdeep Jaitly, Jason Ramapuram, Dan Busbridge, Tatiana Likhomanenko, Joseph Yitan Cheng, Walter Talbott, Chen Huang, Hanlin Goh, and Joshua Susskind. Position prediction as an effective pretraining strategy. 2022. 2, 4, 5, 6

[70] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR, year=2022*. 1, 8

[71] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 7

[72] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1

[73] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2, 7, 8

[74] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, 2022. 1, 2, 3, 7