PartDistillation: Learning Parts from Instance Segmentation

Jang Hyun Cho*
UT Austin
janghyuncho7@utexas.edu

Philipp Krähenbühl
UT Austin
philkr@cs.utexas.edu

Vignesh Ramanathan
Meta AI
vigneshr@meta.com

Figure 1. PartDistillation distills a grouping of embedding distances of the penultimate later of a instance segmentation model into a part segmentation model in a completely unsupervised manner. Above are some representative results obtained **without any part supervision**.

Abstract

We present a scalable framework to learn part segmentation from object instance labels. State-of-the-art instance segmentation models contain a surprising amount of part information. However, much of this information is hidden from plain view. For each object instance, the part information is noisy, inconsistent, and incomplete. PartDistillation *transfers the part information of an instance segmentation model into a part segmentation model through self-supervised self-training on a large dataset. The resulting segmentation model is robust, accurate, and generalizes well. We evaluate the model on various part segmentation datasets. Our model outperforms supervised part segmentation in zero-shot generalization performance by a large margin. Our model outperforms when finetuned on target datasets compared to supervised counterpart and other baselines especially in few-shot regime. Finally, our model provides a wider coverage of rare parts when evaluated over 10K object classes. Code is at* https://github.com/facebookresearch/PartDistillation.

*This work was done during Jang Hyun Cho's internship at Meta AI.

1. Introduction

The world of object parts is rich, diverse, and plentiful. Yet, even the most successful part segmentation benchmarks [10, 22] focus on only the few most prominent image classes, and are orders of magnitude smaller than corresponding object instance segmentation benchmarks [21,31]. Parts are harder to detect, annotate, and properly define.

In this paper, we show that instance segmentation models, and indirectly much larger instance segmentation datasets, provide plentiful supervision for part segmentation. Specifically, we show that the penultimate layer of a pre-trained instance segmentation model readily groups parts across a wide class of instances. We distill this part information from an instance segmentation model into a dedicated part segmentation framework, in a two stage process we call *PartDistillation*. In the first stage, our model learns to segment all possible parts in a class-agnostic fashion. We bootstrap an iterative self-training process from clustered embeddings of an instance segmentation model. The self-supervised nature of this process allows us to scale part discovery to $10K$ object classes in $10M$ images without any part-level supervision. In the second stage, our method

learns to group the discovered parts of each object category independently into object-specific part clusters. Figure 1 shows the result of this two-stage process.

Unlike traditional self-training methods [39, 43, 47] that rely on supervised part labels, we distill the part information from a pre-trained instance segmentation model. In this framework, self-training increases the consistency between the different potential part segmentation, and boosts the noisy supervisory signal. Our model makes full use of powerful instance segmentation architectures [11,12,25] for both supervision and part segmentation itself.

We show that *PartDistillation* outperforms existing unsupervised methods by a large margin. It is very label-efficient in few-shot training, even compared to supervised models trained on existing labelled part segmentation dataset. Finally, we verify that the part discovery quality is consistent beyond a narrow set of classes in existing datasets. We go through manual evaluation process and show that 1) PartDistillation discover more consistent parts compared to supervised model and 2) the precision stays the same when scaled to 10K classes.

2. Related Work

Self-supervised learning aims to learn a general feature representation for many downstream vision tasks by solving a *proxy task* such as instance discrimination [8,9,24,42] and image reconstruction [1, 23]. The learned representation is then finetuned either on the same dataset with few labels or on different datasets and tasks. Other methods directly solve a task without labels such as k-NN classification [5, 42], image retrieval [3, 4], and image segmentation [5, 14, 29, 40]. PartDistillation directly solves part segmentation; we show strong zero-shot performance on unknown datasets and highly label-efficient when fine-tuned.

Unsupervised part segmentation. Some prior works tackles part segmentation in purely unsupervised setting [15,28, 33]. They use a discriminative model to minimize pixel-level contrastive loss and an equivariance loss across views to assign unique labels on different part regions. These models work best if training images contain a single object category centered in an image, and thus do not scale gracefully. In contrast, PartDistillation learns part segmentation from instance segmentation. It uses *object-level* masks and *region-level* representation similar to [2, 11, 25]. In considers features exclusively within a detected instance, enabling the model to learn directly from crowded and scene-centric in-the-wild images.

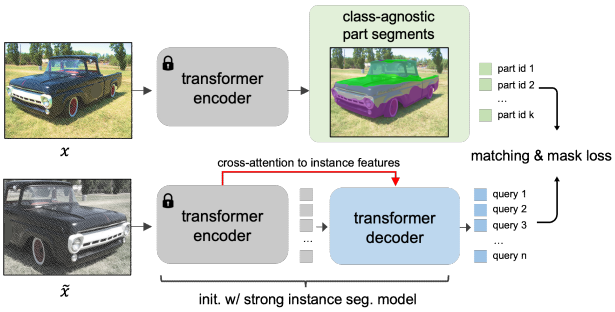
Self-training boosts the performance of a pre-trained model on large-scale unlabelled data. Self-training starts with an initial model trained either with a small portion of labelled data or from self-supervision. It then train another model that predicts the same output as the initial model from a strongly augmented input. This may significantly improve the robustness and performance of the resulting model [20,39,43,47]. PartDistillation can be best described as a self-training method. One notable difference is that we supplement the initial annotated labels with generically mined localization derived from pixel-level feature grouping within each object mask. We show that features from a model trained to solve *object instance segmentation* has surprisingly accurate part-level information which a simple grouping algorithm is able to extract.

Query-based detection and segmentation. Detection Transformer (DETR) [2] rephrases the problem of object detection as a query-based cross-attention mechanism. A set of queries is transformed into *object-level* representation as a single vector by attending to the feature map of a given image through transformer decoders. PartDistillation adapt this framework [2, 11, 12, 46] and learn to represent *part* with a set of queries. This allows to decouple localization and classification over two different stages of training.

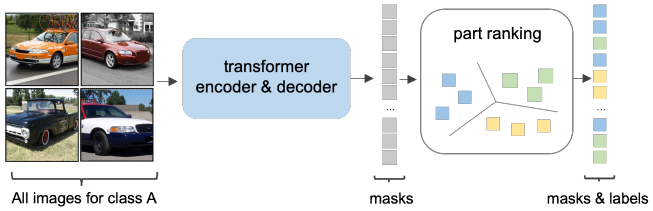
3. Preliminaries

Self-training considers a small set of images and their labels, and a large set of unlabeled images. It starts from a teacher model pre-trained on the available labeled data. The initial model uses a supervised training objective. Self-training then fits a separate student model to the combined supervised data and a corpus of unsupervised data. On the supervised data, it uses the same supervised loss. On the unsupervised data, it uses the signal from the teacher, while heavily augmenting the students inputs. During training, the teacher is periodically updated from a snapshot of the student model. Without such an update self-training closely resembles model distillation [13, 27]. Variants of self-training [20, 41, 47] pre-train on a different task or use self-supervision. Self-training leads to more discriminative and robust features for the final system [3, 4, 20, 39, 47].

Query-based segmentation. Mask2Former is one of the recent methods that introduced the idea of query-based representation of (object instance) segments in an image [11, 12]. Mask2Former starts by encoding an input image I into an intermediate feature representation F using an encoder network $E : I \rightarrow F$. From this feature representation, Mask2Former transforms a fixed set of object queries (learned parameter vectors) $q^o_1, \ldots, q^o_{N^o}$ into object instance masks $M^o_1, \ldots, M^o_{N^o}$ with corresponding objectness scores $s^o_1, \ldots, s^o_{N^o}$ through a decoder network $D^o : q^o_i, F \rightarrow M^o_i, s^o_i, f^o_i$. In addition, each output is associated with a feature vector f^o_i, an abstract representation of the object instance. Mask2Former uses this feature vector to classify objects $c^o_i \in \mathcal{C}$ into pre-defined object categories $\mathcal{C}$ through a classification head. Our PartDistillation makes full use of the query-based Mask2Former. However, instead of producing object instance queries, we produce queries for each potential object part in an image. In the next sec-



(a) First stage: *Part-proposal learning*



(b) Second stage: *Part ranking*

Figure 2. Overview of PartDistillation. **Left**: In the first stage, a transformer encoder produces instance segmentation feature which we group into class-agnostic part segments, *part proposals*, as described in Sec. 4.2. We then train a separate transformer decoder bootstrapped from these part segments and improved through self-training. **Right**: In the second stage, we assign part labels for all part-regions in a class by clustering across dataset and ranking by the density estimates of the clusters. We call this process *class-specific part ranking*.

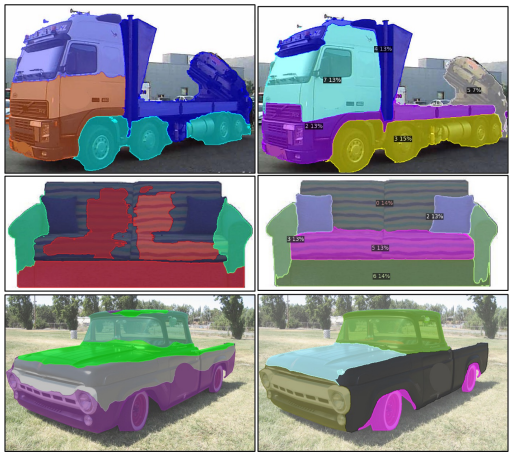


Figure 3. Self-training not only improves localization but also discovers new parts. **Left**: clustered part regions. **Right**: Final PartDistillation prediction after self-training.

tion, we show how to use a variant of self-training, called PartDistillation, to train a query-based segmentation model for object parts without using any part annotations.

4. PartDistillation

Our PartDistillation architecture extends a standard instance segmentation model [11]. We learn an additional query-based part proposal decoder, and an object-class-specific ranking function for each part proposal. Sec. 4.1 presents the exact architecture used for part segmentation. Sec. 4.2 highlights the training objective of the part proposal mechanism, while Sec. 4.3 shows the training of the object-class-specific ranking. Both part proposal and object-class-specific ranking are learned from instance labels alone, and do not use any dedicated part labels. We base all our experiments on a Mask2Former model trained using the open-vocabulary Detic [45] model. See Fig. 2 for an overview.

4.1. A transformer-based part segmentation model

The basic PartDistillation architecture closely follows a Mask2Former [11] object instance segmentation model. We start from a pre-trained instance segmentation model with a fixed encoder E, and object instance decoder D^o. We use both as is and do not further fine-tune or modify them. Instead, we learn a separate part decoder $D^p : q_i^p, F \rightarrow M_i^p, s_i^p, f_i^p$ for a set of generic part queries $q_1^p, \ldots, q_{N^p}^p$. For each part query, we produce a part mask M_i^p, a score s_i^p, and feature representation f_i^p. Here, the score s_i^p highlights how likely an output mask corresponds to a valid part. At this stage, parts are not associated to individual instances, or object classes. Instead, they are shared among all instances and classes in an image. This helps keep the number of potential part queries low, and allows parts to generalize among different object classes.

In a second stage, we assign each part proposal to their closest object instance, and rescore the part in the context of the objects category. For each part query q_i^p, we measure the overlap (Intersection over Union) $\mathcal{O}$ between the part mask M_i^p and all objects masks M_j^o and assign the part to the highest overlapping object a_i:

$$a_i = \arg\max_j \mathcal{O}(M_j^o, M_i^p). \quad (1)$$

Here we use open-vocabulary Detic model to cover large number of object classes. This association provides us with not just an object query, but also its object instance feature $f_{a_i}^o$. We rerank each part proposal using a scoring function $r(f_i^p | f_{a_i}^o)$. We use an object's class as the primary signal to rank part segmentations. Parts that often appear in specific object classes are likely part of an object. Parts that rarely appear in specific object classes may simply be outliers. The final part score $\hat{s}_i^p = r(f_i^p | f_{a_i}^o)$ relies fully on the reranked model.

The final part segmentation model closely resembles two-stage object detection and instance segmentation net-

works. The first stage produces class-agnostic object proposals. A second stage then scores these proposals. The main difference between our setup and two-stage detectors is the training pipeline. In the next two sections, we show how to learn both the part decoder D^p and reranking function $r(f_i^p|f_j^o)$ from just object instance level annotations.

4.2. Learning a part decoder from instance segmentation

We exploit two different signals to train a part segmentation model from a pre-trained instance segmentation: First, within each detected instance, the pixel-level feature representation f^o of an instance segmentation model naturally groups pixels of similar parts together. Second, across a dataset, various parts reoccur, shared between different objects and instances.

PartDistillation starts by clustering pixel-level features of the penultimate layer of the Mask2Former architecture. Given an object instance mask M_i^o, we group pixel-level features f_i^o within that mask by K-Means clustering [36] and obtain class-agnostic part segments $\hat{M}_1^p, \hat{M}_2^p, \ldots, \hat{M}_k^p$ for each object instance. We refer to these segments as *part proposals*. For each object instance, these part proposals follow the inferred instance mask $\hat{M}_j^p \subseteq M_i^o$, and the embedding distance of the mid-level representation f^o of the Mask2Former. The emergence of structured mid-level representations is common among deep networks [44], and as such provides good part-level supervision. However, the resulting grouping is both inconsistent between object instances, and noisy within each instance.

We infer a consistent part segmentation by training a class-agnostic query-based part decoder [11] D^p on all part proposals. More precisely, we train a single-class instance segmentation model and treat all mined part proposals as ground truth masks. We train the model with a binary classification loss and mask loss similar to the original Mask2Former. However, the initial part proposals from pixel-level feature clustering exhibit significant localization errors as visualized in Fig. 3.

Self-training. We reduce this noise through self-training, and obtain high quality *part proposals*. The output of the model is a set of part proposals for each image and the model's confidence scores for the proposals. Additionally, we also obtain decoded query vectors for the proposals, which serve as our *part-level representation*. We filter out part proposals that do not overlap with the object instance mask M^o in each image or have low score s^p. Self-training reinforces positive part proposals, and suppresses the score s^p of negative proposals. The results are clean object-agnostic part proposals, as shown in Fig. 3. In the next section, we show how to assign these proposals to individual object classes, to obtain a list of likely object parts.

Dataset Name	# Images	# Object Classes	# Part Classes (Avg. #)
PartImageNet-*train*	16,540	109	40 ($\sim$ 4)
PartImageNet-*val,test*	7,555	49	40 ($\sim$ 4)
Pascal Part-*train*	4,638	20	50 ($\sim$ 8)
Pascal Part-*val*	4,758	20	50 ($\sim$ 8)
Cityscapes Part-*train*	2,975	5	9 ($\sim$ 4)
Cityscapes Part-*val*	500	5	9 ($\sim$ 4)
ImageNet-21K	$\sim$ 15M	$\sim$ 21K	n/a

Table 1. Summary of all datasets used in this work.

4.3. Learning class-specific part ranking

Our aim is to produce a score $r(f_i^p|f_j^o)$ of how likely a query q_i^p is part of an object q_j^o. We learn this score as a density estimate

$$r_k(f_i^p|f_j^o) = \frac{\exp(-\|D^p(f_i^p, f_j^o) - \mu_k^j\|_2)}{\sum_{l=1}^{N_j} \exp(-\|D^p(f_i^p, f_j^o) - \mu_l^j\|_2)} \quad (2)$$

where the above softmax considers all parts l assigned to an object j. We use an objects class c^o as the main supervisory signal for the above density estimate. Parts that often co-occur with a specific object class in our training set are scored higher, parts that rarely appear in an object category are weighted down. We found a simple weighted k-means-based initial density estimate to be sufficient [38].

During training, we match each pixel x of an object to its most confident $\arg\max_i M_i^p(x)s_i^p$ part query. Any query with a score $s_i^p > 0.3$ that covers at least 5% of the area of the object is considered a candidate. More details about postprocessing part candidates in supplementary. For each class, we aggregate all candidate queries across the entire training dataset. Next, we use the k-means-based density estimator of Snell *et al*. [38] to initialize our scoring function Eq. 2. This density estimate assigns common query features a high initial score, and rare ones a low score. The entire procedure again, does not use any part labels, but instead uses the co-occurrence of parts and object classes over an entire dataset as a supervisory signal.

Final self training. Similar to part proposals, we again use self-training to boost the performance of the class-specific ranking function, with cluster IDs as class labels. We use the same postprocessing step (area and score thresholds) to refine the pseudo-labels before self-training. More details are in the supplementary.

5. Experiments

Datasets. For quantitative evaluation, we use PartImageNet [22], Pascal Parts [10], and Cityscapes Parts [17, 37] datasets. Table 1 shows a summary of these datasets. PartImageNet is a 158 class subset of ImageNet-1K dataset [18] with 40 part classes shared across all object categories. The `test` split of PartImageNet used for evaluation has 49 object categories. All 40 part categories are

	NMI										ARI									
	sheep	horse	cow	mbike	plane	bus	car	bike	dog	cat	sheep	horse	cow	mbike	plane	bus	car	bike	dog	cat
DFF	12.2	14.4	12.7	19.1	16.4	13.5	9.0	17.8	14.8	18.0	21.6	32.3	23.3	37.2	38.3	28.5	24.1	39.1	32.3	37.5
SCOPS	26.5	29.4	28.8	35.4	35.1	35.7	33.6	28.9	30.1	33.7	46.3	55.7	51.2	59.2	68.0	66.0	67.1	52.4	52.2	46.6
K-means	34.5	33.3	33.0	38.9	42.8	37.5	38.4	35.2	40.4	44.2	58.3	66.8	59.0	63.1	76.8	66.4	70.6	63.2	70.2	71.9
Choudhury et al.	35.0	37.4	35.3	40.5	45.1	38.8	36.8	34.8	46.6	47.9	59.8	68.9	59.7	64.7	79.6	67.6	72.7	64.7	73.6	75.4
Choudhury et al.†	55.2	42.8	60.3	**42.5**	49.4	45.1	41.1	39.8	**51.2**	**55.4**	77.4	62.8	81.8	**61.5**	70.9	74.1	66.4	54.2	**86.2**	**88.9**
PartDistillation	**57.3**	**62.2**	**65.5**	34.8	**58.8**	**55.6**	**54.8**	**53.6**	43.6	37.8	**81.6**	**89.9**	**90.0**	43.7	**88.7**	**84.1**	**87.5**	**74.8**	59.0	52.6

Table 2. We evaluate our *single* model predictions on all 10 individual models of DFF [16], SCOPS [28], and Choudhury et al. [15]. We follow the same evaluation protocol as [15] such as the number of parts, image resizing and cropping, etc. Note that our model has never seen Pascal Part images. Here † means our implementation with comparable model.

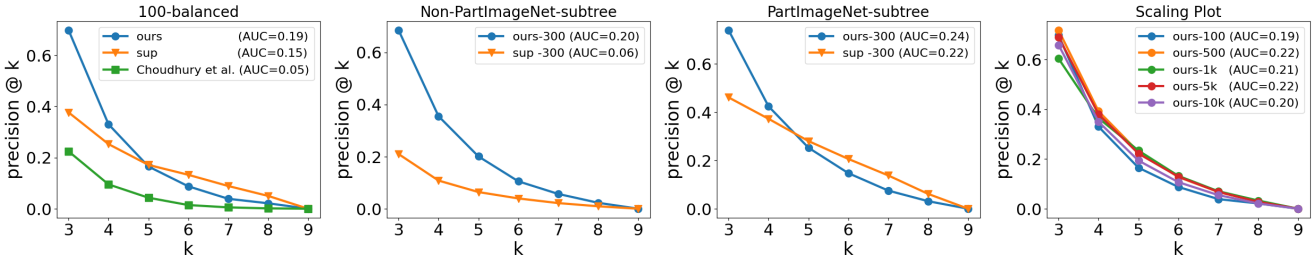


Figure 4. Manual evaluation result comparing supervised method and PartDistillation.

still present in the `test` split. Pascal Parts is another part segmentation dataset with 20 object categories and 50 overall part categories. Each object class has 9 part classes on average. Cityscapes Parts contains 5 object classes that are either *person* or *vehicles*. Interestingly, there are a few common object classes between all datasets. However, each dataset has different definitions of parts for those objects. While all our models are only trained on ImageNet-21K with $15M$ images, we also compare with baseline models trained directly on the `train` split of above evaluation datasets. We evaluate all models on the `val,test` splits of the evaluation datasets.

Baselines. Here we describe all the baselines. In addition to published methods for unsupervised part segmentation like Choudhury et al. [15], we also describe some simple variants of our approach like "one-stage self-training". We also compare with fully supervised models in specific settings. All models are Mask2Former [11] with SwinL backbone [34], initialized with weights trained on COCO Instance Segmentation [32] unless otherwise specified.

(1) Choudhury et al. segment parts by training a model for a single object class at a time. Hence, we train individual models, one class at a time for every dataset. We resort to only using the DeepLab [7]-like framework with SwinL backbone as suggested in their work.

(2) One-stage self-training. We also extend the standard one-stage unsupervised segmentation by clustering with self-training. In particular, we first use K-Means to cluster pixels belonging to all segmented object instances of each object category. We use the part clusters as initial supervisory signal to run two rounds of self-training.

(3) Part-supervised models. We evaluate models trained with full supervision using a source dataset on a new target dataset. This allows us to compare their generalization ability with our fully unsupervised model.

Implementation Details. Self-training in our model is done with a batch size of 256 over 4 nodes. We use a learning rate of 0.0001 except for fine-tuning experiments in Table 4. For part-proposal learning step, we trained our model for $50K$ iterations, and $100K$ iterations for the final self-training during part association. For training Choudhury et al. [15], we closely followed their official implementation. When training on PartImageNet/ImageNet, we chose the best hyper-parameter based on one randomly chosen object category and used it for all other categories. For part-region mining, we set $k = 4$ and for association we set $k = 8$. We applied dense-CRF [30] for each mined parts offline. We provide more details are in supplementary.

5.1. Evaluation on annotated part datasets

We first compare our method against baseline models on PartImageNet, PascalParts and Cityscapes Parts which have pre-annotated part masks for different object categories. Unsupervised methods (such as our method) associate segmented parts with arbitrary cluster labels. Unlike supervised methods, there is no direct one-to-one correspondence between cluster labels and pre-annotated part labels.

NMI and ARI. The metrics normalized mutual information (NMI) and adjusted Rand index (ARI) were introduced in Choudhury et al. [15] as a way to cope with the above issue. NMI and ARI measure quality with respect to the target annotated part mask labels.

Mean Intersection over Union (mIoU). Another standard way to evaluate unsupervised methods is to first associate each generated cluster with one of the part labels in the dataset whose part masks have the highest mIoU overlap

Figure 5. An example of 3×3 grid images shown to annotators for a part cluster generated by our method (object class: laptop). The discovered parts are highlighted in red.

with the segments from the cluster. This allows us to directly adopt mIoU to compare the cluster with the masks from the associated ground truth part. We provide more details in supplementary.

Average Recall. We also evaluate only the localization ability of the model separately. This can be done with Average Recall (AR@k) metric [32] for measuring quality of the top k predicted masks per image. The metric ignores class labels and measures recall@k at different IoU thresholds, followed by averaging across the thresholds.

5.2. Purity evaluation through manual rating

Metrics used for unsupervised segmentation evaluation have different trade-offs as pointed out in previous works [15]. Also, existing part datasets only cover a small set of object categories, and expanding part annotations to thousands of objects in ImageNet is prohibitive in terms of annotation cost. Hence, to measure the quality of discovered parts at a much larger scale, we need a different strategy. Inspired by both observations, we introduce a manual rating based metric. More specifically, for each part cluster discovered for an object category, we sample 9 images to form a 3×3 grid and highlight the part mask belonging to the cluster as shown in Fig. 5. We ask annotators to rate the purity on a scale of 3-9, denoting the maximum number of images in the grid, where the part mask consistently segments the same part of the object. As an example, Fig. 5 was rated as 9. The higher the rating, the purer the cluster. This rating process is very cheap, since even for $10K$ classes with 8 parts, we need to rate only $80K$ collages. This does not require annotators to manually draw part boundaries, which is much more expensive.

5.3. Results

Comparison with unsupervised methods. We provide results for various unsupervised methods on PascalParts in Tab. 2. We follow the same evaluation protocol as the state-of-the-art method from Choudhury et al. [15], and use NMI and ARI metrics. We compare with published results from [15] for DFF [16], SCOPS [28] and K-Means. These methods use a Resnet50 backbone. Extending them methods to a transformer based framework is beyond the scope of this work (details in supp.). However, to enable fair comparison, we also train the best baseline: Choudhury et al. [15] with our implementation using comparable models (SwinL) and pre-trained weights from Mask2Former.

We further note that our method is trained only on ImageNet-21K dataset and is not fine-tuned on PascalParts[1]. Despite this, it outperforms the SwinL version of Choudhury et al. trained on PascalParts for 7 out of 10 object classes. Our NMI averaged across all classes is 52.4 which is 4.1% better than that for Choudhurty et al. (48.3). This shows the higher quality of parts discovered by our method, in addition to its ability to easily scale to $10K$ object classes unlike existing methods.

Comparison with supervised methods. The primary focus of our work is to scale part segmentation to a large number of object classes. This could also be achieved if supervised part models trained on a small source dataset with part annotations, can be transferred to new target datasets. We refer to this as zero-shot part segmentation. We compare PartDistillation with supervised models in this zero-shot setup in Tab. 3. We train the supervised model on different source datasets and evaluate on a new target dataset. Note that our method is still trained without any part annotations. We

Method	Train	Test	GT Object	AR@200	mIOU
Supervised	PartImageNet	Pascal Part		10.6	
Supervised	Cityscapes Part	Pascal Part		12.3	
One-stage.	ImageNet-21K	Pascal Part		18.8	
PartDistillation	ImageNet-21K	Pascal Part		**25.0**	
Supervised	PartImageNet	Pascal Part	✓	11.1	21.8
Supervised	Cityscapes Part	Pascal Part	✓	13.4	12.0
One-stage.	ImageNet-21K	Pascal Part	✓	20.4	16.5
PartDistillation	ImageNet-21K	Pascal Part	✓	**26.8**	**23.0**
Supervised	Pascal Part	PartImageNet		40.8	30.8
Supervised	Cityscapes Part	PartImageNet		10.8	17.2
One-stage.	ImageNet-21K	PartImageNet		31.8	26.6
PartDistillation	ImageNet-21K	PartImageNet		**51.4**	**36.1**
Supervised	Pascal Part	PartImageNet	✓	45.7	34.9
Supervised	Cityscapes Part	PartImageNet	✓	11.9	21.5
One-stage.	ImageNet-21K	PartImageNet	✓	36.2	31.9
PartDistillation	ImageNet-21K	PartImageNet	✓	**58.0**	**48.0**

Table 3. Zero-shot part segmentation comparing PartDistillation to supervised baselines. PartImageNet and Pascal Part consist of object classes that share the common part classes. The zero-shot part segmentation is measured by AR@200 and mIOU. Despite that supervised models are trained with part labels, PartDistillation consistently shows better generalization. "One-stage." stands for our one-stage self-training baseline.

[1] One advantage that PartDistillation enjoys is learning parts from many object classes, hence easily scalable to large dataset. Other unsupervised baselines are specifically designed for a single object class, and with our best effort we could not successfully train on multi-class extension of any of the baselines. Hence, other baselines are trained on one class at a time.

Method	Train	Finetune	AR@200						mIOU					
			1 %	5%	10%	20%	50%	100%	1 %	5%	10%	20%	50%	100%
Instance Seg.	COCO	Pascal Part	25.3	36.5	38.8	43.6	47.1	49.4	20.5	37.3	44.4	52.6	55.5	56.0
Part-supervised	PartImageNet	Pascal Part	28.7	37.9	40.6	42.9	46.0	48.5	23.0	**44.2**	**51.4**	52.7	55.6	56.3
Part-supervised	Cityscapes Part	Pascal Part	25.1	35.3	39.1	41.3	44.6	46.5	17.0	34.6	43.1	51.5	54.8	54.9
PartDistillation	ImageNet-21K	Pascal Part	**33.2**	**40.5**	**42.4**	**45.5**	**47.8**	**50.2**	**25.8**	43.0	48.7	**53.0**	**56.2**	**58.6**
Instance Seg.	COCO	PartImageNet	65.3	70.5	73.5	75.8	76.3	76.8	32.0	58.3	63.2	66.1	68.9	70.8
Part-supervised	Pascal Part	PartImageNet	59.5	65.8	67.0	71.0	73.6	76.6	**45.6**	59.5	63.4	65.9	68.2	69.5
Part-supervised	Cityscapes Part	PartImageNet	52.5	63.4	67.7	69.6	73.2	73.2	20.6	54.0	60.5	65.5	67.3	70.0
PartDistillation	ImageNet-21K	PartImageNet	**67.8**	**73.2**	**74.1**	**76.1**	**77.3**	**76.9**	36.3	**60.3**	**64.5**	**67.2**	**70.0**	**71.5**
Instance Seg.	COCO	Cityscapes Part	12.5	17.4	18.8	**20.3**	20.3	21.2	28.3	49.9	51.3	61.5	62.4	63.8
Part-supervised	PartImageNet	Cityscapes Part	8.4	13.7	15.0	17.8	18.1	18.7	37.1	50.9	55.1	62.6	64.1	66.2
Part-supervised	Pascal Part	Cityscapes Part	14.1	16.6	17.6	18.8	19.3	19.7	53.0	62.3	63.3	65.7	67.6	68.4
PartDistillation	ImageNet-21K	Cityscapes Part	**14.7**	**19.1**	**19.2**	**20.3**	**20.7**	**21.3**	**53.4**	**63.1**	**64.4**	**66.2**	**68.7**	**69.9**

Table 4. Few-shot benchmark. We train PartDistillation and part-supervised models on 1%, 5%, 10%, 20%, 50% and 100% of the target data with labels and evaluate AR@200 and mIOU to measure part segmentation. All models are initialized as Mask2Former with SwinL backbone pretrained on COCO instance segmentation dataset for 100 epochs. For "Part-supervised" and PartDistillation, numbers in gray cells are initialized from part-proposal models as explained in Sec. 4.2.

use two setups during evaluation, where we assume ground truth object masks are available in the test set or not.

We first evaluate AR@200 and observe that our method significantly outperforms all supervised methods in all settings by at-least 10%. The gap is much higher ($\geq 15\%$), when measured on PascalParts which has a more diverse set of part and object classes. This shows the superior localization ability of our method compared to even models trained with explicit part mask supervision. We also see that our full method outperforms the one-stage self-training baseline, validating the need for the two-stage design.

Since part labels in target datasets are different from those in source datasets, we ignore the predicted part labels from the supervised models and only retain the part masks. We then treat these masks as "part proposals", similar to the output from the first stage of our model, and cluster proposals belonging to instances of each object category in the `test` split of the target dataset (details in supplementary). This can be done easily in the case of PartsImageNet even in the absence of ground truth object masks, since each image mostly has only one instance of one object category. However, this is not possible for PascalParts which often has multiple object categories and instances per image. Hence, we report mIoU on PascalParts only when ground truth object masks are assumed to be available. We notice significant gains from our model on the mIoU metric as well compared to other methods, including the one-stage baseline.

Few-shot part segmentation. In practice, another way to scale part segmentation would be to collect a small amount (few-shot) of part mask annotations for the target dataset and object categories. We could then start from a strong pretrained model and fine-tune it on the target few-shot dataset. We evaluate in such a setup as well to further measure the practical utility of our model. We compare different pretraining methods in Tab. 4. This includes our method as well as part-supervised models trained with full part supervision on different source datasets. Additionally, we also compare with a model pre-trained for COCO instance segmentation. We report performance with different amounts of supervision (1%-100%) for the target dataset.

We first notice that at a very low-shot setup (1%) our model outperforms all other methods significantly. In the case of Pascal parts, it is 4.5% better is AR@200 than the next best method (33.2 vs 28.7). More interestingly, our method achieves an mIoU of 56.2% with only 50% labels. This is comparable to the best performance with other methods at 100% supervision of 56.0. This demonstrates that our method is $2\times$ more label efficient than other methods. This observation is true for other target datasets as well.

5.4. Purity evaluation with manual rating

We first randomly sample 100 object classes from ImageNet-21K dataset and compare our method with both Choudhury et al.[†] (trained with our SwinL implementation) as well as the supervised model trained on PartImageNet. As before, for the supervised method we discard part labels from the source dataset and cluster the resulting class-agnostic part masks independently for each of the 100 object categories to get object-specific part clusters. We use Detic [45] to obtain instance segmentation mask for the object category in each image, and restrict part masks to be fully contained within the instance mask for all methods. We obtain 8 part clusters per object category for all methods. We eliminate degenerate parts that occupy a large portion of the object, by setting a threshold for the part mask area. We remove part masks whose area exceed 50% of the object area (more analysis in supplementary).

The clusters were sent for manual rating as mentioned in Sec. 5.2. The ratings can range from 3-9. In Fig. 4 at a given purity rating r, we plot the total fraction of parts (denoted as precision) across all object categories that were annotated with a rating $\geq r$. The higher the value the bet-

Feature	Dataset	Pretrain Task	Framework	AR@1	AR@10
ResNet-101 [26]	IN-21K	Classification	n/a	2.8	6.4
ConvNeXt-L [35]	IN-21K	Classification	n/a	3.1	7.9
ViT-L	IN-1K	DINO [5]	n/a	2.2	5.6
ViT-L	IN-1K	MAE [23]	n/a	1.6	4.6
ViT-L	COCO	MAE+Inst. Seg.	HTC++	2.8	7.3
SwinL	IN-21K	Classification	n/a	5.3	9.4
SwinL	COCO	Inst. Seg.	HTC++	2.9	12.5
SwinL	LVIS	Inst. Seg.	Mask2Former	5.3	23.1
SwinL	LVIS + COCO	Inst. Seg.	Mask2Former	5.5	24.3
SwinL	COCO	Inst. Seg.	Mask2Former	**6.6**	**27.6**

Table 5. Part localization of pixel-grouping from different architectures, pretrain task, and framework measured by AR@1 and AR@10 on PartImageNet-`test`. Here we compare ResNet [26], ConvNext [35], ViT [19], and Swin transformer [34] as candidates.

Multi-scale	Cosine	Instance-level	AR@1	AR@10
			1.2	9.8
	✓		2.1	16.1
		✓	3.5	14.4
	✓	✓	5.5	23.4
✓		✓	5.2	21.5
✓	✓	✓	**6.6**	**27.6**

Table 6. Ablating different features, distance metric, and clustering scope on part localization performance measured by AR@1 and AR@10 on PartImageNet-`test`.

ter the clustering purity of the method. In Fig. 4.a we notice that both our method and the supervised method outperform Choudhury et al. Our performance is better than the supervised model. However, we noticed the gap between the curves to be small. This is due to a large fraction of the 100 sampled classes belonging to the same WordNet subtree as those used in PartImageNet (anmials and vehicles). To test the hypothesis, we sampled 300 classes outside the PartImageNet subtree and 300 classes inside this subtree to do another round of manual evaluation, reported in Fig. 4.b and Fig. 4.c respectively. We notice that the performance from our method generalizes equally well in both settings, but the supervised model does not generalize outside PartImagenet subtree. We provide non-cherry-picked visualization for many object classes in the supplementary.

Next, we also measure the scaling ability of our model in Fig. 4.d, where we compare the performance of our model on differing number of object classes in ImageNet, ranging from 100 to 10K. We notice that the performance of our method is consistent, irrespective of the number of object classes being evaluated.

5.5. Ablations

We conduct ablations to study the factors that influence the localization of initial mined-part segments. We use the ground truth instance mask to select pixel-level features on each object instance and measure the localization quality by AR@1 and 10 on the `test` split of PartImageNet dataset.

Choice of architecture. We evaluate the mined parts from different settings. We use K-Means to mine part regions as before and measure the AR@1 and AR@10 with different settings. We observe that the localization quality differs significantly, and Swin Transformer [34] trained with

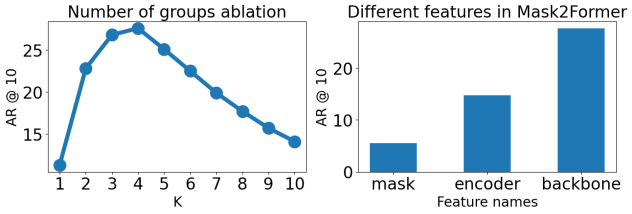


Figure 6. Different ablations for the pixel grouping and its localization quality. **Left:** Different number of groups per instance measured with AR@10. **Right:** Features from different stage of Mask2Former measured with AR@10. Both are evaluated on *val* and *test* combined splits of PartImageNet [22].

Mask2Former [11] on COCO instance segmentation task shows a significant improvement compared to all other choices in Tab. 5. Interestingly, even with the same pretraining task, the choice of model is crucial. Mask2Former has an AR@10 of 27.6 compared to only 12.5 from a model trained for the same task but with HTC++ [6].

Choice of features. In Tab. 6, we keep the backbone fixed and explore the optimal choice of features (multi-scale combining res3 and res4 layers vs only using res4) from Mask2Former and the distance metric (cosine vs L2) for clustering. We also show the impact of clustering pixels from all instances of the dataset together (in line with standard one-stage models [14] for unsupervised segmentation) vs our "instance-level" clustering where we cluster pixels from one object instance at a time. We note that cosine similarity yields better results in general.

Fig. 6 (right) shows the localization quality with features from different layers in Mask2Former [11]. "mask", "encoder", and "backbone" stand for mask features, transformer encoder features, and backbone features all from Mask2Former, respectively.

Number of clusters. In Fig. 6 (left), we measure the localization of the mined part segments from Sec. 4.2 with different numbers of grouping, and we pick the optimal $k = 4$.

Instace-level vs dataset-level clustering. In Tab. 6, instance-level segmentation in conjunction with cosine similarity provides AR@10 of 23.4 which is much higher than the equivalent one-stage setting without instance-level segmentation 16.1. We also note that multi-scale features provide a significant gain as well 23.4 vs 27.6.

6. Conclusion and Discussion

In this work, we present a scalable self-training pipeline that can learn part segmentation for 21K object classes without any part segmentation labels. We show that PartDistillation has strong performance in zero-shot and few-shot settings. We show that the discovered parts have consistent purity over 10K part classes by manual evaluation.

Acknowledgments. This material is in part based upon work supported by the National Science Foundation under Grant No. IIS-1845485 and IIS-2006820.

References

[1] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020. 2

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[4] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2, 8

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 8

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 5

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[10] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 1, 4

[11] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 2, 3, 4, 5, 8

[12] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 2

[13] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019. 2

[14] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *CVPR*, 2021. 2, 8

[15] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems*, 34:28104–28118, 2021. 2, 5, 6

[16] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 5, 6

[17] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 4

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 8

[20] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021. 2

[21] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 1

[22] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. *arXiv preprint arXiv:2112.00933*, 2021. 1, 4, 8

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2, 8

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[27] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2

[28] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–878, 2019. 2, 5, 6

[29] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019. 2

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

[31] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 1

[32] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 6

[33] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8355–8364, 2021. 2

[34] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 5, 8

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[36] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 4

[37] Panagiotis Meletis, Xiaoxiao Wen, Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Cityscapes-panoptic-parts and pascal-panoptic-parts datasets for scene understanding. 4

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 4

[39] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. In *arXiv:2005.04757*, 2020. 2

[40] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10052–10062, October 2021. 2

[41] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. FreeSOLO: Learning to segment objects without annotations. *arXiv preprint arXiv:2202.12181*, 2022. 2

[42] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2

[43] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[44] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4

[45] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022. 3, 7

[46] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2

[47] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 2