**Abstract**

Recent self-supervised pretraining methods for object detection largely focus on pretraining the backbone of the object detector, neglecting key parts of detection architecture. Instead, we introduce DETReg, a new self-supervised method that pretrains the entire object detection network, including the object localization and embedding components. During pretraining, DETReg predicts object localizations to match the localizations from an unsupervised region proposal generator and simultaneously aligns the corresponding feature embeddings with embeddings from a self-supervised image encoder. We implement DETReg using the DETR family of detectors and show that it improves over competitive baselines when finetuned on COCO, PASCAL VOC, and Airbus Ship benchmarks. In low-data regimes DETReg achieves improved performance, e.g., when training with only 1% of the labels and in the few-shot learning settings.¹

¹Code: [https://www.amirbar.net/detreg/](https://www.amirbar.net/detreg/).

**1. Introduction**

Object detection is a key task in computer vision, yet it largely relies on the availability of human-annotated training datasets. Building such datasets is not only costly but sometimes infeasible for privacy-sensitive applications such as medical imaging or personal photos [46, 68]. Fortunately, recent advancements in self-supervised representation learning have substantially reduced the amount of labeled data needed for a variety of applications, including object detection [6, 11, 26, 27].

Despite this recent progress, current approaches are limited in their ability to learn good representations for object detection because they do not pretrain the entire object detection network, specifically the localization and region embedding components. Most recent works (e.g., SwAV [6], ReSim [62], InsLoc [67]) follow the same pretraining playbook for the detection network as a supervised image-classification-based pretraining, where only the CNN backbone can be initialized from the pretrained model. While the recent UP-DETR [16] method pretrains a full detection architecture, it still does not localize objects within the image, but rather random image regions.

In this work, we present a model for Detection with Transformers using Region priors (DETReg), which unlike existing pretraining methods, learns to both localize and encode objects simultaneously in the unsupervised pretraining stage – see Figure 1. DETReg involves two object-centric and
category-agnostic pretraining tasks: an Object Localization Task to localize objects, and an Object Embedding Task to encode an object’s visual properties. Taken together, these tasks pretrain the entire detection network – see Figure 2 for an overview. A final object classification head can then be finetuned with a small number of labels yielding better performance than existing methods.

DETReg’s object localization task uses simple region proposal methods for class-agnostic bounding-box supervision [3, 3, 14, 15, 54]. These methods require little or no training data and can produce region proposals at a high recall. For example, Selective Search [54], the region proposal method we adopt in DETReg, uses object cues such as continuity in color, hierarchy, and edges to extract object proposals. DETReg builds upon these region priors to learn a class-agnostic detector during pretraining.

DETReg’s object embedding task aims to predict the embeddings of a separate self-supervised image encoder evaluated on object regions. Self-supervised image encoders, e.g., SwAV [6], learn transformation-invariant embeddings, so training the detector to predict these values distills the learned invariances into the detector’s embeddings. Thus, the object embedding head learns representations that are robust to transformations such as translation or image cropping.

We conduct an extensive evaluation of DETReg on standard object detection benchmarks like MS COCO [41] and PASCAL VOC [18], and on an aerial images dataset, Airbus Ship Detection [1]. We find that DETReg improves the performance using two state-of-the-art base architectures compared to challenging baselines, especially when small amounts of annotated data are available.

Quantitatively, DETReg improves over a backbone-only image-classification pretraining baseline by 4 AP points on PASCAL VOC, 1.6 AP points on MS COCO, and 1.2 AP points on Airbus Ship Detection. Additionally, DETReg outperforms pretraining baselines in semi-supervised learning when using 1% to 10% of data, and on 10 and 30 shot. Taken together, these results indicate that pretraining an entire detection network, including region proposal prediction and embedding components, is beneficial and that our specific DETReg model realizes new SOTA performance by taking advantage of this object-centric self-supervised pretraining.

2. Related Work

Self-supervised pretraining. Recent work [6, 10, 13, 22, 25, 27, 28, 31, 43] has shown that self-supervised pretraining can generate powerful representations for transfer learning, even outperforming its supervised counterparts on challenging vision benchmarks [10, 61]. The learned representations transfer well to image classification but the improvement is less significant for instance-level tasks, such as object detection and instance segmentation [27, 29, 48, 70].

More recently, a number of works [31, 51, 62, 64] focused on learning backbones that can transfer to object detection. In contrast to these works, we pretrain the entire detection network. As we show, pretraining the backbone with an image-patch-based task does not necessarily empower the model to learn what and where an object is, and adding weak supervision from the region priors proves beneficial.

Our approach is also different from semi-supervised object detection approaches [34, 42, 53, 65] and few-shot detection approaches [9, 12, 19, 20, 35, 36, 39, 56, 59, 60, 63, 65, 69] as we initialize the detector from a pretrained DETReg model without further modifying the architecture. Therefore, these approaches can be viewed as complementary to DETReg.

End-to-end object detection. Detection with transformers (DETR) [5] builds the first fully end-to-end object de-
tector and eliminates the need for components such as an-
chor generation and non-maximum suppression (NMS) post-
processing. This model has quickly gained traction in the
machine vision community. However, the original DETR
suffers from slow convergence and limited sample efficiency.
Deformable DETR [71] introduced a deformable attention
module to attend to a sparsely sampled small set of promi-
lient key elements, and achieved better performance com-
pared to DETR with reduced training epochs. We therefore
use Deformable DETR as our base detection architecture.

Both DETR and Deformable DETR adopt the super-
vised pretrained backbone (ResNet [30]) on ImageNet. UP-
DETR [16] pretrains DETR in a self-supervised way by
detecting and reconstructing the random patches from the
input image. Instead, we additionally adopt region priors from
unsupervised region proposal algorithms to provide weak
supervision for pretraining, which has an explicit notion of
objects rather than the random patches used by UP-DETR.

**Region proposals.** A rich study of region proposals meth-
ods exists in the object detection literature [2, 3, 8, 15, 17,
37, 55, 72]. Grouping based method, Selective Search [55],
and window scoring based approach, Objectness [2] are two
early and well known proposal methods, which have been
widely adopted and supported in major software libraries
(e.g., OpenCV [4]). Selective Search greedily merges super-
pixels to generate proposals. Objectness relies on visual cues
such as multi-scale saliency, color contrast, edge density and
superpixel straddling to identify likely regions.

While the field has largely shifted to learning-based ap-
proaches, the key benefit of these models is that they require
little or no training data, and can produce region proposals
at a high recall [3, 3, 14, 15, 54]. This provides a cheap, albeit
noisy, source of supervision. Hosang et al. [32, 33] offer a
comprehensive analysis over the various region proposals
methods, and Selective Search is among the top perform-
ing approaches in terms of recall. Here, we seek weak su-
ervision from the region proposals generated by Selective
Search, which has been widely adopted and proven success-
ful in the well-known detectors such as R-CNN [24] and
Fast R-CNN [23]. However, our approach is not limited to
Selective Search and can employ other proposal methods.

3. DETReg

DETRReg is a self-supervised method to fully pretrain
object detectors, including their region localization and em-
bedding components. At a high level, DETReg operates by
predicting object localizations that match those from an un-
supervised region proposal generator, while simultaneously
aligning the corresponding feature embeddings with embed-
dings from a self-supervised image encoder, see Figure 2.

The key idea underlying DETReg is to formulate pretext tasks that are similar to the tasks performed during super-
vised object detection, so that improved pretraining trans-
fers to the object detector. We built DETReg based on the
DETR family of detectors [5, 71] due to their implementation
simplicity and performance, though other architectures can
easily plug into DETReg. Next, we review DETR, and in
the following subsections, we present the object localization
and embedding pretext tasks that form the core of DETReg.

**DETR summary:** DETReg detects up to \( N \) objects in an
image by iteratively applying attention and feed-forward lay-
ers over \( N \) object query vectors of a transformer decoder and
over the input image features. The last layer of the decoder
results in \( N \) image-dependent query embeddings that are
used to predict bounding box coordinates and object cate-
gories. Formally, consider an input image \( x \in \mathbb{R}^{H \times W \times 3} \).
DETR uses \( x \) to calculate \( N \) image-dependent query embed-
dings \( v_1, \ldots, v_N \) with \( v_i \in \mathbb{R}^d \). This is achieved by passing
the image through a backbone, followed by a transformer,
and processing of the query vectors [5]. Then, two predic-
tion heads are applied to \( v_i \). The first, \( f_{\text{box}} : \mathbb{R}^d \rightarrow \mathbb{R}^4 \),
predicts the bounding boxes. The second, \( f_{\text{cat}} : \mathbb{R}^d \rightarrow \mathbb{R}^L \),
outputs a distribution over \( L \) object categories, including a
background “no object” category.

3.1. Object Localization Task

DETRReg’s object localization pretraining task uses simple
region proposal methods for class-agnostic bounding-box
supervision (see the orange arrows in Figure 2). We use
the output from these methods as they require limited or
no training data and can produce region proposals at a high
recall [3, 14, 15, 54]. We use Selective Search [54] as the
primary region proposal method for training DETReg as it
is widely available in off-the-shelf computer vision libraries
and requires no training data. Selective Search uses object
cues such as continuity in color and edges to extract object
proposals, and DETReg further builds upon these region
priors to learn a class-agnostic detector.

Region proposal methods take an image and produce a
large set of region proposals at a high recall rate, where some
of the regions are likely to contain objects. However, they
have low precision and do not output category information,
see [32, 33]. Since the content of non-object boxes tends to
be more variable than of object boxes, we expect that deep
models can be trained to recognize the visual properties of
objects even when given noisy labels.

Thus, the **Object Localization** pretraining task takes a
set of \( M \) boxes \( b_1, \ldots, b_M \) (where \( b_i \in \mathbb{R}^d \)) output by
an unsupervised region proposal method and optimizes a
loss that minimizes the difference between the detector box
predictions (the output of the \( f_{\text{box}} \), MLP) and these \( M \) boxes.
Similar to DETR, the loss involves matching the predicted
boxes and these \( M \) boxes, a process we detail in Section 3.3.

Common region proposal methods attempt to sort the
region proposals such that proposals that are more likely to
be objects appear first, however, the number of proposals is typically large, and the ranking is not precise. Therefore, we explore methods to choose the best regions to use during training. We consider three policies for selecting boxes:

Top-K uses the top-

Random-K uses K random proposals, which may yield lower quality proposals but encourages exploration.

Importance Sampling relies on the region proposal method ranking but also encourages more diverse proposals. Formally, let \( b_1, \ldots, b_n \) be a set of \( n \) sorted region proposals, where the \( b_i \) has rank \( i \). Let \( X_i \) be a random variable indicating whether we will output the \( b_i \). We assign the sampling probability for \( X_i \) to be: \( Pr(X_i = 1) \propto -\log(i/n) \). To determine if a box should be included, we randomly sample from its respective distribution.

3.2. Object Embedding Task

In the supervised training of object detectors, every box is associated with a class category of the object, which is not available in an unsupervised setting. Therefore, to learn a strong object embedding, we encode each box region \( b_i \) via a separate encoder network and obtain embeddings \( z_i \) that are used as a target for the DETReg embeddings \( \hat{z}_i \) (see the black arrows in Figure 2).

The separate encoding network that produces \( z_i \) could be jointly trained by following similar bootstrapping techniques from works such as BYOL [26] or DINO [7]. However, for training stability and to reduce the convergence time, we leverage a pretrained, self-supervised model whose embeddings are invariant to many image transformations, e.g., blurring and color distortions. Here we primarily use a SwAV [6] pretrained model as it is one of the strongest performing methods for pretraining image classifiers and has readily available code and pretrained models.

To predict a corresponding object embedding \( \hat{z}_i \) in the detector, we introduce an additional MLP \( f_{emb} : \mathbb{R}^d \rightarrow \mathbb{R}^d \) that predicts the object embedding \( \hat{z}_i \) from the corresponding DETR query embedding, \( v_i \). This encourages \( v_i \) to capture the information that is useful for category prediction. The loss is the \( L_1 \) loss between \( z_i \) and \( \hat{z}_i \).

3.3. DETReg Pretraining

Here, we formally describe how DETReg optimizes the localization and embedding tasks during pretraining. Assume that our region proposal method returns \( M \) object proposals which are used to generate \( M \) bounding boxes \( b_i \) and object descriptors \( z_i \) for \( i \in \{1, \ldots, M\} \), and let \( y_i = (b_i, z_i) \) with \( y = \{y_i\}_{i=1}^M \). DETReg is trained such that its \( N \) outputs align with \( y \).

Let \( v_1, \ldots, v_K \) denote the image-dependent query embeddings calculated by DETR (i.e., the output of the last layer of the DETR decoder). DETReg has three prediction heads: \( f_{box} \) which outputs predicted bounding boxes, \( f_{cat} \) which predicts if the box is object or background, and \( f_{emb} \) which reconstructs the object embedding descriptor. Denote these outputs as: \( \hat{b}_i = f_{box}(v_i) \), \( \hat{z}_i = f_{emb}(v_i) \), \( \hat{p}_i = f_{cat}(v_i) \), and define \( \hat{y}_i = (\hat{b}_i, \hat{z}_i, \hat{p}_i) \) and \( \hat{y} = \{\hat{y}_i\}_{i=1}^N \).

Following DETR training, we assume that the number of DETR queries \( N \) is larger than \( M \), so we pad \( y \) to obtain \( N \) tuples, and assign a label \( c_i \in \{0, 1\} \) to each box in \( y \) to indicate whether it is a region proposal (\( c_i = 1 \)) or padded proposal (\( c_i = 0 \)); see the green arrows in Figure 2. With the DETR family of detectors [5, 71], there are no assumptions on the order of the labels or the predictions and therefore we first match the objects of \( y \) to the ones in \( \hat{y} \) via the Hungarian bipartite matching algorithm [38]. Specifically, we find the permutation \( \sigma \) that minimizes the optimal matching cost between \( y \) and \( \hat{y} \):

\[
\sigma = \arg\min_{\sigma \in \Sigma_N} \sum_{i=1}^N L_{match}(y_i, \hat{y}_{\sigma(i)}) \tag{1}
\]

Where \( L_{match} \) is the pairwise matching cost matrix as defined in [5, 71] and \( \Sigma_N \) is the set of all permutations over \( \{1 \ldots N\} \). Using the optimal \( \sigma \), we define the loss as:

\[
L(y, \hat{y}) = \sum_{i=1}^N \lambda_{class} L_{class}(c_i, \hat{p}_{\sigma(i)}) + \lambda_{box} L_{box}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{emb} L_{emb}(z_i, \hat{z}_{\sigma(i)}) \tag{2}
\]

Where \( L_{class} \) is the class loss, that can be implemented via Cross Entropy Loss or Focal Loss [40], and \( L_{box} \) is based on the the \( L_1 \) loss and the Generalized Intersection Over Union (GloU) loss [50]. Finally, we define \( L_{emb} \) to be the \( L_1 \) loss:

\[
L_{emb}(z_i, \hat{z}_j) = \|z_i - \hat{z}_j\|_1 \tag{3}
\]

4. Experiments

We first describe the implementation details and datasets used for our experimentation. We then report how DETReg performs on the object detection tasks when fine-tuned on full and low-data regimes, including few-shot learning, and semi-supervised learning. Finally, we conclude with ablations, analyses, and visualizations from DETReg.

Implementation. Based on the ablations presented in Section 4.5, the default experiment settings are as follows (see the Suppl. for all details). We initialize the ResNet50 backbone of DETReg with SwAV [6], which was pretrained with multi-crop views for 800 epochs on IN1K, and fix it throughout the pretraining stage. In the object embedding branch, \( f_{emb} \) and \( f_{box} \) are MLPs with 2 hidden layers of size 256 followed by a ReLU [44] nonlinearity. The output sizes of \( f_{emb} \) and \( f_{box} \) are 512 and 4. \( f_{cat} \) is implemented as a single fully-connected layer with 2 outputs. Unless otherwise
Results. Table 1 shows that DETReg consistently outperforms other pretraining strategies using both DETR and Deformable DETR. For example, DETReg improves the COCO AP score by 1.4 points compared to UP-DETR when trained for 150 epochs, and in fact, outperforms the 300 epoch supervised variant after 80 epochs. Interestingly, using DETReg pretraining with DETR is competitive with supervised Deformable DETR, which achieves only 0.8 points (AP) more, despite significant architectural modifications.

Table 2 shows that DETReg improves by 2.5 (AP) points over SwAV on PASCAL VOC and by 1.2 (AP) on Airbus. For reference, by using a specialized architecture for ship detection that builds on a ResNet50 backbone, as well as leveraging the pixel-level annotations, [45] obtains a box AP score of 76.1 on this dataset, 4.9 points lower than DETReg, which only uses the bounding box annotations.

4.2. Object Detection in Low-Data Regimes

These experiments test how DETReg performs when a small amount of annotated data is available for finetuning.

Table 1. Object detection results when trained on MS COCO train2017 and evaluated on val2017. Both DETReg and UP-DETR are pretrained on IN1K under comparable settings, while supervised and SwAV only pretrain the backbone of the object detector. We explore both the DETR and Deformable DETR (DDETR) architectures; for compatibility with prior work, we fine-tuned the DETR for 150/300 epochs and DDETR for 50 epochs.

| Method  | PASCAL VOC | Airbus Ship |
|---------|------------|-------------|
|        | AP | AP$_{50}$ | AP$_{75}$ | AP | AP$_{50}$ | AP$_{75}$ |
| Supervised | 59.5 | 82.6 | 65.6 | 79.8 | 95.8 | 89.4 |
| SwAV [6] | 61.0 | 83.0 | 68.1 | 78.3 | 95.7 | 88.7 |
| DETReg | 63.5 | 83.3 | 70.3 | 81.0 | 95.9 | 89.7 |

Table 2. Object detection finetuned on PASCAL VOC and Airbus Ship data. The model is finetuned on PASCAL VOC trainval07+2012 and evaluated on test07 (left), and Airbus Ship Detection finetuned on the train split and evaluated on the 3k test images (right). All models are based on Deformable DETR [71]. Bold values indicate an improvement $\geq 0.3$ AP.

4.1. Object Detection in Full Data Regimes

These experiments test how well DETReg performs when a fully annotated dataset is available for finetuning.

Pretraining. We pretrain two variants of DETReg based on DETR [5] and Deformable DETR [71] detectors for 5 and 60 epochs on IN1K and IN100, respectively, where the pretraining schedules are set by proportionally adjusting the schedules used in UP-DETR to equate to the more efficient Deformable DETR schedules [71].

Baselines. We compare DETReg to several closely related state-of-the-art pretraining approaches for object detection with transformers: using a SwAV [6] backbone, a fully pre-trained UP-DETR [16], and a supervised baseline backbone.

Experiments. To evaluate DETReg, we finetuned it on three different datasets: MS COCO [41], PASCAL VOC [18], and Airbus Ship Detection [1]. We perform an extensive comparison on MS COCO and finetune using similar training schedules as previously reported in [16, 71], using train2017 for finetuning and val2017 for evaluation. On PASCAL VOC and Airbus we use DETReg Deformable DETR based variant, which is faster to train. On PASCAL VOC we finetune on trainval07+12 for 100 epochs, dropping the learning rate after 70 epochs and use the test07 for evaluation. For Airbus, we finetune for 100 epochs, dropping the learning rate after 80 epochs.

Results. Table 1 shows that DETReg consistently outperforms other pretraining strategies using both DETR and Deformable DETR. For example, DETReg improves the COCO AP score by 1.4 points compared to UP-DETR when trained for 150 epochs, and in fact, outperforms the 300 epoch supervised variant after 150 epochs. Interestingly, using DETReg pretraining with DETR is competitive with supervised Deformable DETR, which achieves only 0.8 points (AP) more, despite significant architectural modifications.
We train all methods for up to 2TReg when transferring with limited amounts of labeled data. We consider Deformable DETR with a supervised pretrained backbone as the most direct baseline as validation performance stops improving. Then, we fine-tune on a balanced set of all 80 classes, where the difference in AP compared to the supervised baseline, where the x-axis shows the total number of images used during training. We fix the Deformable DETR architecture across all methods and finetune it using publicly released ResNet50 weights of different methods on MS COCO train2017 and evaluate on val2017.

Results. Figure 3 shows the results, where the y-axis reports the difference in AP compared to the supervised variant. The results indicate that DETReg consistently outperforms other pretraining strategies, when using Deformable DETR on low data regime. For example, when using only 256 images, DETReg improves the average precision (AP) score by 4.1 points compared to 1.1 for SwAV and 0.5 for ResSim.

4.3. Few-Shot Object Detection

These experiments test how DETReg extends to the few-shot settings established in existing literature. Pretraining. We pretrain DETReg based on Deformable DETR [71] for 5 epochs on ImageNet (IN1K).

Baselines. We consider Deformable DETR with a supervised pretrained backbone as the most direct baseline as its architecture and training strategy mirror DETReg. We also report the results of recent few-shot approaches, which utilize different underlying object detectors. Concurrent to our work, Meta-DETR [69] proposed a new method based on Deformable DETR. However, unlike DETReg, it uses a ResNet101 backbone and a single image scale, but we include its results to encourage unified reporting, even when experimental settings are not perfectly aligned.

Experiments. Following the standard few-shot protocol for object detection [56], we finetune DETReg on the full data of 60 base classes, which contain around 99K labeled images. Then, we finetune on a balanced set of all 80 classes, where every class has $k \in \{10, 30\}$ object instances. We use the splits from [56] and report the performance on the novel 20 classes. The results are shown in Table 3.

Table 3 shows that DETReg achieves competitive few-shot performance even when the model is not trained on the abundant base class data. As a reference point, TFA [56] is a previous fine-tuning method that trains on the abundant base class data, and we can see that DETReg outperforms it without additional supervision from the base class data.

4.4. Semi-supervised Learning

These experiments test how DETReg compares to semi-supervised methods, where small amounts of labeled data and large amounts of unlabeled data are used during training. Pretraining. We pretrain DETReg (Deformable DETR) for

![Figure 3. Model comparison in low-data regimes. $\Delta AP$ improvement over the supervised baseline, where the x-axis shows the total number of images used during training. We fix the Deformable DETR architecture across all methods and finetune it using publicly released ResNet50 weights of different methods on MS COCO train2017 and evaluate on val2017.](image-url)
Table 4. Few-shot object detection without training on the COCO base classes. To test DETReg’s performance on extreme few-shot scenarios, we conduct an evaluation where DETReg is finetuned only on the K-shot COCO subsets. DETReg performs slightly worse (∼10%) or better (∼30%) compared to TFA [56] without using base class data while also using a smaller backbone.

Table 5. Object detection using k% of the labeled data on COCO. The models are trained on train2017 using k% and then evaluated on val2017.

Table 6. Ablation studies. This tables ablates region proposal sampling strategies, values of $\lambda_{emb}$, and whether to freeze backbone with DETReg trained on IN100 and finetuned on MS COCO. Shuffling the region proposals across images led to a 11.3 AP drop, $L_{emb}$ has a consistent performance, and freezing the backbone does not significantly change the performance.

Table 7. Class agnostic object proposal evaluation on MS COCO val2017. The models are trained on IN100 and for each method, we consider the top 100 proposals. We show DETReg identifies objects more effectively than the previous methods.

50 epochs on MS COCO train2017 without labels.

Baselines. We compare DETReg with a Deformable DETR model initialized with a supervised backbone from IN1K pretraining, which is the most direct baseline as all experiments are carried out on the same architecture and training data. We consider recent approaches for pretraining ResNet50 backbone for object detection like ResSim [62] and SwAV [6], for each we use the publicly released checkpoint.

Experiments. We finetuned DETReg on random k% of train2017 data for $k \in \{1, 2, 5, 10\}$, until convergence (validation performance stopped improving). In each setting, we train 5 different models with different random seeds and report the mean and standard deviation.

Results. Table 5 shows that DETReg outperforms existing pretraining methods, including a consistent improvement over the supervised pretraining baseline. We include a more broad comparison in the Supplementary Table 9, where we also compare to approaches that leverage both the labeled and unlabeled data via auxiliary losses [34, 42, 53, 65].

4.5. DETReg Analysis

This section further explores and justifies the architectural and algorithmic choices used in the main experiments.

Design Ablations. Table 6 examines the contribution of the object localization and object embedding tasks in DETReg. To quantify the importance of using object-centric region proposals, we train DETReg while randomly shuffling the proposal box locations across images, as indicated by “Shuffle” in the “Proposals” column. Second, to assess the contribution of the embedding loss $L_{emb}$, we evaluate DETReg with different coefficients $\lambda_e \in \{0, 1, 2\}$. Finally, we validate that performance does not drop when freezing the backbone during training, i.e. that the performance benefits stem from the core DETReg contributions. All models are trained on IN100 for 50 epochs and finetuned on MS COCO.

Table 6 justifies our design choices: shuffling the region proposals across images led to a 11.3 AP drop indicating that the object-centric proposal are important. We further see that the embedding loss $L_{emb}$ has a relatively consistent performance improvements with changes of $\leq 2$ AP for all setting, and we select $\lambda_e = 1$ based on these results. Finally, the performance of DETReg with and without freezing the backbone encoder is relatively consistent with changes of 0.3 AP points between the two settings.

Class Agnostic Object Detection. We examine the class agnostic performance of DETReg variants discussed in Section 3, as well as region proposal and pretraining approaches. The results reported in Table 7 indicate that DETReg variants achieve improved performance over other pretraining approaches including solely using Selective Search. This indicates that coupling the object embedding and localization components in the DETReg model improves the localization ability. In addition, we observe that the Top-K region pro-
Figure 4. DETReg visualization. We show the gradient norms from the unsupervised DETReg detection with respect to the input image $I$ for (top) the $x$ coordinate of the object center, (middle) the $y$ coordinate of the object center, (bottom) the feature-space embedding, $z$.

Proposal selection strategy performs best in these ablations.

**Robustness to different proposal methods.** We test how DETReg performs when pretrained with Selective Search proposals compared to Edge Box region proposals [72]. Specifically, we pretrain DETReg on IN100 and finetune on MS COCO with 2% and 10% of random data. We find that both variants perform similarly well with AP of 21.8 vs. 21.0 for 2% and a similar result of 36.2 for 10%.

Visualizing DETReg. Figure 4 shows qualitative examples of DETReg unsupervised box predictions with Deformable DETR. Additionally, it shows the Saliency Map [52] of the $x/y$ bounding box center and the object embedding with respect to the input image $I$. The first three columns show the attention focusing on the object edges for the $x/y$ predictions and $z$ for the predicted object embedding. The final column shows a case where the background plays a more important role than the object in the embedding. We believe this may be due to the CNN-based encoder focusing on the textures rather than the shapes in the region as discussed in [21], and we view further exploration of such characteristics as an intriguing direction for future work.

5. Limitations

DETReg’s localization pretraining task uses simple region proposal methods for class-agnostic bounding-box supervision [3, 14, 15, 54]. While Table 7 indicates that DETReg performance can improve beyond these methods, DETReg class-agnostic results remain far behind supervised counterparts. Furthermore, our experiments focused on DETR [5]-related architectures, but it may be possible that DETReg applies to more traditional detection architectures, which we leave for future work to explore. Finally, while DETReg improves training time, transformer-based object detectors still require significant computational resources to train.

6. Conclusion

We presented DETReg, an unsupervised pretraining approach for object detection with transformers using region priors. Through extensive empirical study, we showed DETReg learns representations in the unsupervised pretraining stage that lead to improvements in downstream performance for two different transformer models across three different datasets and many settings. We believe unsupervised pretraining holds the potential for positive social impact, mainly because it can utilize unlabeled data and reduce the need for massive amounts of labeled data which can be very expensive for fields like Medical Imaging. We do not anticipate a negative impact specific to our approach, but as with any model, we recommend careful validation before deployment.

Acknowledgements

We would like to thank Sayna Ebrahimi and Jitendra Malik for their helpful feedback. This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC Holi 819080). Prof. Darrell’s group was supported in part by DoD including DARPA’s LwLL and/or SemaFor programs, as well as BAIR’s industrial alliance programs. This work was completed in partial fulfillment for the Ph.D degree of the first author.
References

[1] Airbus. airbus ship detection challenge. https://www.kaggle.com/c/airbus-ship-detection. Accessed: 2021-09-30. 2, 5

[2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In CVPR. IEEE, 2010. 3

[3] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In CVPR. 2014. 2, 3, 8

[4] G. Bradski. The OpenCV Library. Dr. Dobb’s Journal of Software Tools, 2000. 3, 12

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. ECCV, 2020. 2, 3, 4, 5, 8, 12, 13

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS, 2020. 1, 2, 4, 5, 7, 12

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jéougu, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294, 2021. 4

[8] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. TPAMI, 34(7):1312–1328, 2011. 3

[9] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018. 2

[10] 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, 5, 12

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. NeurIPS, 2020. 1

[12] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, and Winston H Hsu. Should i look at the head or the tail? dual-awareness attention for few-shot object detection. arXiv preprint arXiv:2102.12152, 2021. 2

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020. 2, 12

[14] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. IEEE transactions on pattern analysis and machine intelligence, 37(3):569–582, 2014. 2, 3, 8

[15] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3286–3293, 2014. 2, 3, 8

[16] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UPTDETR: Unsupervised pre-training for object detection with transformers. CVPR, 2021. 1, 3, 5, 7, 12

[17] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. TPAMI, 36(2):222–234, 2013. 3

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

[19] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4013–4022, 2020. 2

[20] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4527–4536, 2021. 2

[21] Robert Girshick, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231, 2018. 8

[22] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In CVPR, 2020. 2, 12

[23] Ross Girshick. Fast r-cnn. In ICCV, 2015. 3

[24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 3

[25] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In ICCV, 2019. 2, 12

[26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doerﬂer, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheslaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS, 2020. 1, 4

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, 2020. 1, 2, 5

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, 2020. 2, 5, 12

[29] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In ICCV, 2019. 2

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 3

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

[32] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? TPAMI, 38(4):814–830, 2015. 3

[33] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? arXiv preprint arXiv:1406.6962, 2014. 3
end semi-supervised object detection with soft teacher. *arXiv preprint arXiv:2106.09018*, 2021. 2, 7, 12, 13

[66] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9577–9586, 2019. 2, 6

[67] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *CVPR*, 2021. 1, 5

[68] Jun Yu, Baopeng Zhang, Zhengzhong Kuang, Dan Lin, and Jianping Fan. iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Transactions on Information Forensics and Security*, 12(5):1005–1016, 2016. 1

[69] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. Meta-detr: Few-shot object detection via unified image-level meta-learning. *arXiv preprint arXiv:2103.11731*, 2021. 2, 6

[70] Dongzhan Zhou, Xinchi Zhou, Hongwen Zhang, Shuai Yi, and Wanli Ouyang. Cheaper pre-training lunch: An efficient paradigm for object detection. In *European Conference on Computer Vision*, pages 258–274. Springer, 2020. 2

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 3, 4, 5, 6, 12

[72] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. Springer, 2014. 3, 8
Supplementary Material

We start by providing the full implementation details of DETReg and include the complete PASCAL VOC results. We then follow with additional analysis of DETReg pretraining as well as class agnostic performance and visualization.

Implementation Details. Based on the ablations presented in Section 4.5, the default experiment settings are as follows.

For region proposals, we compute Selective Search boxes online using the “fast” preset of the OpenCV implementation [4] and unless otherwise noted, we use the DETReg Top-K region selection variant (see Section 3.1) and set $K = 30$ proposals per-image. We initialize the ResNet50 backbone of DETReg with SwA V [6], which was pretrained with multi-crop views for 800 epochs on IN1K, and fix it throughout the pretraining stage. A similar SwAV encoder is used to encode region proposals, which are first cropped and resized to 128x128. In the object embedding branch, $f_{emb}$ and $f_{box}$ are MLPs with 2 hidden layers of size 256 followed by a ReLU [44] nonlinearity. The output sizes of $f_{emb}$ and $f_{box}$ are 512 and 4. $f_{cat}$ is implemented as a single fully-connected layer with 2 outputs. We run the pretraining experiments using a batch size of 24 per GPU on an NVIDIA DGX, V100 x8 GPUs machine, following the hyperparameter settings and image augmentations from existing works [5, 71]. Similarly, cropped regions are augmented before being fed to the encoder to obtain embeddings $z_i$. When finetuning, we drop the $f_{emb}$ branch, and set the size of the last fully-connected layer of $f_{cat}$ to be the number of classes in the target dataset plus a background class.

Object Detection in Full Data Regimes

We reported DETReg results on the PASCAL VOC benchmark in Section 4.1. Here we include the full table, containing more past pretraining approaches using three different object detectors (see Table 8). We observe that using the Deformable-DETR detector, the supervised pretraining baseline is superior to past pretraining approaches and that DETReg pretraining improves over it by 4 points (AP). (FRCN), DETR [5], and Deformable DETR [71] (DDETR). Bold values indicate an improvement $\geq 3$ AP.

| Method      | Detector | AP  | AP50  | AP75  |
|-------------|----------|-----|-------|-------|
| Supervised  |          | 56.1| 82.6  | 62.7  |
| InsDis [61] |          | 55.2| 80.9  | 61.2  |
| Jigsaw [25] |          | 48.9| 75.1  | 52.9  |
| NPID++ [43] |          | 52.3| 79.1  | 56.9  |
| SimCLR [10] |          | 51.5| 79.4  | 55.6  |
| PIRL [43]   |          | 54.0| 80.7  | 59.7  |
| BoWNet [22] |          | 55.8| 81.3  | 61.1  |
| MoCo [28]   |          | 55.9| 81.5  | 62.6  |
| MoCo-v2 [13]|          | 57.0| 82.4  | 63.6  |
| SwA V [6]   |          | 56.1| 82.6  | 62.7  |
| DenseCl [57] |         | 58.7| 82.8  | 65.2  |
| DetCo [64]  |          | 58.2| 82.7  | 65.0  |
| ReSim [62]  |          | 59.2| 82.9  | 65.9  |
| Supervised  | DETR     | 54.1| 78.0  | 58.3  |
| UP-DETR [16]|          | 57.2| 80.1  | 62.0  |
| Supervised  | DDETR    | 59.5| 82.6  | 65.6  |
| SwA V [6]   |          | 61.0| 83.0  | 68.1  |
| DETReg      |          | 63.5| 83.3  | 70.3  |

Table 8. Object detection finetuned on PASCAL VOC. The model is finetuned on PASCAL VOC trainval07+2012 and evaluated on test07. Models are based on Faster-RCNN [49] (FRCN), DETR [5], and Deformable DETR [71] (DDETR). Bold values indicate an improvement $\geq 0.3$ AP.

Semi-supervised Learning

We reported DETReg results and comparisons to other pretraining approaches like [6, 62] when using limited amounts of data. In Table 9, we include comparisons to semi-supervised works [34, 42, 53, 65] that leverage both the labeled and unlabeled data in training via auxiliary losses.

DETReg Analysis

In Section 4.5 we analyzed DETReg, including the model ablations, class agnostic results, visualization and robustness. Here we further examine the pretrained DETReg model including the class agnostic results, and TopK selection policy.

Figure 5. Top-K proposals performance of Selective Search. Using different values of $K$, we evaluate the class agnostic performance of Selective Search on MS COCO 2017 validation split.

Improved Encoder, improved DETReg. We test how DETReg performs when object embeddings are obtained with different image encoders. Specifically, we pretrain DETReg on IN100 using SwA V trained for 400 epochs compared to a superior variant trained for 800 epochs with multi-crops. We finetune on MS COCO with 1% data and observe the improved encoder achieves 1 AP improvement (27.7 vs 26.7).

DETReg TopK selection policy. Using Selective Search, we examine the class agnostic performance when using TopK...
Table 9. **Object detection using k% of the labeled data on COCO.** The models are trained on train2017 using k% and then evaluated on val2017. Methods like [42] utilize auxiliary losses during the training stage using unlabeled data, whereas DETReg utilizes unlabeled data during the pretraining stage only.

| Method     | Approach | Detector | COCO 1%       | COCO 2%       | COCO 5%       | COCO 10%      |
|------------|----------|----------|---------------|---------------|---------------|---------------|
| CSD [34]   |          |          | 10.5 ± 0.1    | 13.9 ± 0.1    | 18.6 ± 0.1    | 22.5 ± 0.1    |
| STAC [53]  | Auxiliary| FRCN     | 14.0 ± 0.6    | 18.3 ± 0.3    | 24.4 ± 0.1    | 28.6 ± 0.2    |
| U-T [42]   |          |          | 20.8 ± 0.1    | 24.3 ± 0.1    | 28.3 ± 0.1    | 31.5 ± 0.1    |
| S-T [65]   |          |          | **20.5 ± 0.4**| --            | **30.7 ± 0.1**| **34.0 ± 0.1**|
| Supervised |          |          | 11.31 ± 0.3   | 15.22 ± 0.32  | 21.33 ± 0.2   | 26.34 ± 0.1   |
| SwAV       | Pretraining| DDETR  | 11.79 ± 0.3   | 16.02 ± 0.4   | 22.81 ± 0.3   | 27.79 ± 0.2   |
| ReSim      |          |          | 11.07 ± 0.4   | 15.26 ± 0.26  | 21.48 ± 0.1   | 26.56 ± 0.3   |
| DETReg     |          |          | **14.58 ± 0.3**| **18.69 ± 0.2**| **24.80 ± 0.2**| **29.12 ± 0.2**|

Figure 6. **DETRreg slots specialize in specific areas in the image and uses a variety of box sizes much like Deformable DETR.** Each square corresponds to a DETR slot, and shows the location of its bounding box predictions. We compare 10 random slots of the supervised Deformable DETR (top) and unsupervised DETReg (bottom) decoder for the MS COCO 2017 val dataset. Each point shows the center coordinate of the predicted bounding box, where following a similar plot in [5], a green point represents a square bounding box, an orange point is a large horizontal bounding box, and a blue point is a large vertical bounding box. Deformable DETR has been trained on MS COCO 2017 data, while DETReg has only been trained on unlabeled ImageNet data. Similar DETReg and Deformable DETR slots were manually chosen for illustration.

Policy. We report the precision and recall in Figure 5. In this paper, we have used $K = 30$ (see Figure 7), which emphasizes precision over recall. This might imply that DETReg performs well given high precision proposals.

DETRreg Slots Visualization. We examine the learned object queries slots (see Figure 6) and observe they are similar to those in Deformable DETR, despite not using any human annotated data. Nevertheless, the Deformable DETR slots have greater variance with respect to locations and they tend to specialize more in particular boxes shapes.

Class Agnostic Object Detection. The quantitative results in Section 4.5 indicate that DETReg improves over Selective Search. The included qualitative examples of DETReg on MS COCO (see Figure 8) support a similar conclusion, indicating that DETReg outperforms Selective Search but still much behind the ground truth labeled data.

Figure 7. **TopK Selective Search proposals on ImageNet.** Using K=30, the proposals typically cover objects and parts-of-objects in the image.
Figure 8. **Class Agnostic object detection visualization.** Examples predictions using Selective Search and DETReg on random MS COCO images. For every image annotated with $M$ boxes, only the top $M$ predictions are shown.