Region Proposal Network Pre-Training Helps Label-Efficient Object Detection

Linus Ericsson[1*†], **Nanqing Dong**[2*], **Yongxin Yang**[3], **Aleš Leonardis**[3], **Steven McDonagh**[3]
[1]University of Edinburgh [2]University of Oxford
[3]Huawei Noah's Ark Lab
linus.ericsson@ed.ac.uk
nanqing.dong@cs.ox.ac.uk
{yongxin.yang,ales.leonardis,steven.mcdonagh}@huawei.com

Abstract

Self-supervised pre-training based on instance discrimination has fueled the recent advance in label-efficient object detection. However, existing studies focus on pre-training only a feature extractor network to learn transferable representations for downstream detection tasks. This leads to the necessity of training multiple detection-specific modules from scratch in the fine-tuning phase. We argue that the region proposal network (RPN), a common detection-specific module, can additionally be pre-trained towards reducing the localization error of multi-stage detectors. In this work, we propose a simple pretext task that provides an effective pre-training for the RPN, towards efficiently improving downstream object detection performance. We evaluate the efficacy of our approach on benchmark object detection tasks, including autonomous driving and common objects. In comparison with multi-stage detectors without RPN pre-training, our approach is able to consistently improve downstream task performance, with largest gains found in label-scarce settings.

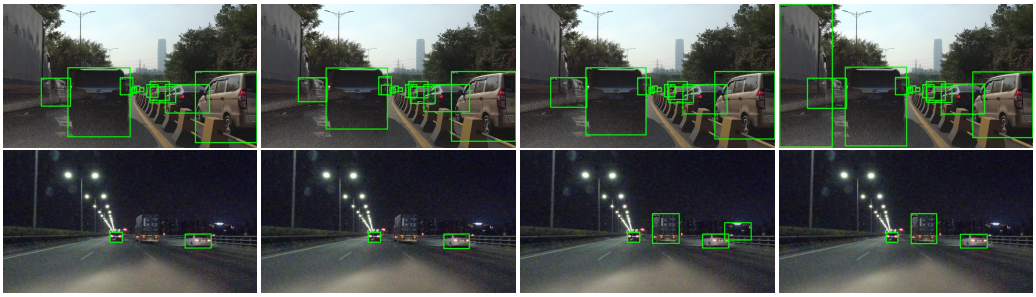
1 Introduction

Image-level representation learning based on instance discrimination has proven highly effective as a general model for transfer learning, successfully transferring to diverse downstream tasks [17, 14, 4, 9]. Object detection poses a fundamental and challenging computer vision problem, and solving it successfully requires combination of (1) finding object locations, and (2) classifying located objects. Popular contemporary supervised object detection commonly tackles this by employing multi-stage architectures that suggest object proposals (locations) which a sequential network must then recognise. In contrast, most existing self-supervised pre-training for object detection tasks focus solely on solving component (2); through approximation of a classification loss [26, 27, 5, 25]. This leads to situations where classification-related architectural components receive gradient updates from both unlabeled (pre-training) and labeled (fine-tuning) data, while localization-related components receive gradient updates only from labeled data. We conjecture that such training setups are sub-optimal for overall performance, especially in cases offering only limited labeled data.

A promising recent design principle involves encouraging alignment between pretext task and downstream task. SoCo [25] achieves state-of-the-art performance by pre-training model components consisting of feature extractor and detector head components. Our insight into the importance of localization-based errors motivate us to extend these pre-training concepts, to further evaluate whether model components based on region proposal networks (RPN) [19] can provide meaningful performance improve-

[*]The first two authors contributed equally at Huawei Noah's Ark Lab.
[†]Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) BYOL (b) SwAV (c) SoCo (d) ADePT (Ours)

Figure 1: Qualitative comparison on the SODA10M dataset [15]. Top: All models successfully detect the cars on the right hand side but our ADePT model is the **only** one to find the truck on the left hand side. Bottom: Dark background and low illumination make successful detection challenging. BYOL [14] and SwAV [3] fail to detect all three vehicles. While SoCo [25] and ADePT both successfully capture **all** three vehicles with high confidence, SoCo hallucinates an object on the right hand side. **Best viewed with digital zoom**.

ment, particularly in situations with few labelled instances. Directly addressing the aforementioned localization error issues has, until now, remained largely missing from self-supervised approaches.

Towards this we develop a fully aligned self-supervised pre-training approach for object detection, named **A**ligned **De**tection **P**re-**T**raining (ADePT). In contrast to supervised detection training setups, ADePT does not require: (a) ground-truth class labels; classification is approximated by treating cropped regions as individual classes and solving an instance discrimination task, and further does not require: (b) ground-truth bounding boxes; we will show that location labels can be successfully approximated via unsupervised region proposals, generated by heuristic algorithms. We instantiate the latter idea in this work using selective search [21].

Through evaluation of our ideas, we aim to answer the following research questions: **Q1.** Can the inclusion of RPN module pre-training reduce (commonly high) localization-based detection error terms? **Q2.** Does principled alignment of pretext and downstream tasks improve overall performance? **Q3.** Does such alignment enable and benefit more label-efficient scenarios?

We evaluate ADePT on the recent autonomous driving benchmark SODA10M [15] and the standard detection task MS COCO [18]. We find that ADePT consistently provides stronger downstream detection performance over several seminal self-supervised pre-training baselines, especially in domains containing only limited labeled data (*e.g.* when only 1% and 10% of labels are available).

2 Method

In this section, we focus on the details for our novel RPN pre-training strategy, the first contribution of this work, and defer remaining architectural model details to Appendix.

We follow common two-stage object detector architectural design that consists of three main architectural components: a feature extractor, an RPN and a detector head. The training of the feature extractor and detector head follows a BYOL-style contrastive learning strategy [14], where two networks with identical architectures are denoted *online* and *target* networks, respectively. In the training phase, coarse bounding boxes are first generated by selective search [21]. The input image and generated bounding boxes are then used to generate three augmented views (augmentation details provided in Appendix B). These views are passed through the feature extractor and the resulting features are fed to the RPN to generate region proposals. In this way we uniquely train the RPN to regress the selective search bounding boxes. The refined features are then extracted from the learned proposal regions using "RoIAlign" [16]. Analogous to BYOL, the refined features are fed through the detector head, projector, predictor and subsequently contrasted in the detector loss which follows the SoCo formulation and is defined in Appendix B. The total loss is a combination of the RPN and detector losses, together pre-training the entire architecture. The online network receives gradient updates while the target network is maintained as the exponential moving average (EMA) of the online network. In the inference phase, only the target branch is kept and the prediction is made by feeding an input image

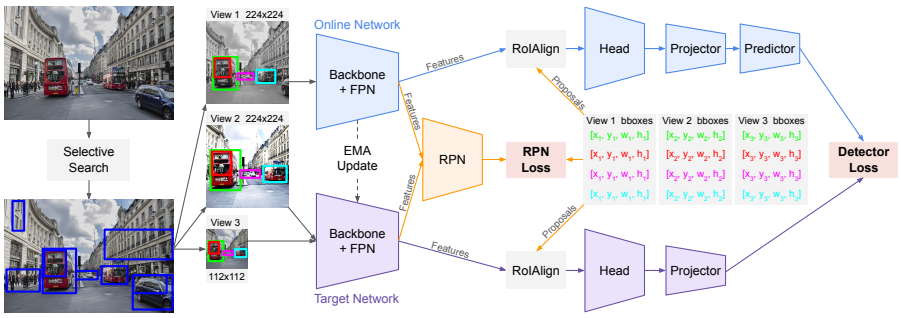


Figure 2: Diagram of ADePT. Our proposed contribution involves self-supervised pre-training of the RPN component. The feature extractor (backbone + FPN) and detector head are also trained in self-supervised fashion where three augmented views of the same region in the original image are passed through an online branch (blue) and a target branch (purple), respectively to form a contrastive detector loss. Further pipeline details are found in Appendix B.

through the target feature network, the RPN, and the target detector head sequentially. Our complete ADePT pipeline is illustrated in Fig. 2. Further RPN training details can be found in the Appendix.

3 Experiments

We aim to evaluate pre-trained models in terms of the quality of their transferable representations. Firstly, following previous work [25], we consider the autonomous driving dataset SODA10M [15], along with detection of common objects in MS COCO [18]. In addition to the results presented in the main paper, the Appendix also presents results for few-shot object detection, a challenging small-scale data setting, as well as ablation studies to better understand credit assignment and constituent component attributions.

Experimental Setup: Our experimental setting follows [25]. For fair comparison, all SSL methods are pre-trained using ImageNet [7] under identical settings. We examine the transferablity of learned representations by fine-tuning pre-trained weights on multiple downstream tasks. We fine-tune with stochastic gradient descent and a batch size of 16 split across 8 GPUs.

We compare our proposed approach with recent alternative SSL methods. Among the considered baselines, three representative baselines[3] are BYOL [14], a seminal SSL baseline based on an online-target encoder architecture and image-level contrastive loss; SwAV [3] based on contrastive learning and clustering; SoCo [25], the state-of-the-art. Both SoCo and ADePT leverage a BYOL-style training strategy in terms of the online-target encoders setup and instance-wise discrimination pretext task (contrastive learning). SwAV serves as an alternative contrastive learning baseline in contrast to BYOL. In addition, we quote further common baselines reported in previous work [25, 15], such as MoCo [17], SimCLR [6], DenseCL [24], and DetCo [26].

We first consider the **SODA10M** dataset [15] which consists of driving scenes. The dataset contains six object classes with bounding box annotations, namely: {car, truck, pedestrian, tram, cyclist, tricycle} where the training set consists of only 5000 images captured from a single city in clear weather, during the day. We train a Cascade R-CNN object detector [2] with ResNet50-FPN backbone for 3726 iterations on the full (labeled) training set of 5000 images. We then evaluate using the 5000 image validation set where varying weather conditions, cities and periods of the day are present, making for a challenging object detection test with potential domain shifts.

Next, we consider the **MS COCO** dataset [18] and use the `train2017` subset, containing ∼118k images for training. We train a Mask R-CNN [16] detector with a ResNet50-FPN backbone for 90000 iterations. Eighty object categories are annotated with both bounding box and instance segmentation labels, allowing evaluation of the object detection and instance segmentation tasks. Further, we simulate two *label-scarce* settings under this dataset and additionally train using 10% and 1% of the training set respectively, for 9000 iterations in each case. The transfer learning performance of the

[3]For these three baseline SSL methods, the results are achieved by directly running the given source code under the corresponding experimental setups for fair comparison.

Pre-training			SODA10M	COCO		COCO 10%		COCO 1%	
Method	Architecture	Epochs	AP_{bb}	AP_{bb}	AP_{mk}	AP_{bb}	AP_{mk}	AP_{bb}	AP_{mk}
Scratch	R50-FPN	-	25.4	26.4	44.0	-	-	-	-
Supervised	R50-FPN	90	32.9	38.2	35.4	-	-	-	-
MoCo [17]	R50-FPN	200	32.3	38.5	35.1	-	-	-	-
SimCLR [6]	R50-FPN	200	32.8	-	-	-	-	-	-
SwAV [3]	R50-FPN	200	33.9	38.5	35.4	-	-	-	-
BYOL [14]	R50-FPN	300	-	40.4	37.2	-	-	-	-
DenseCL [24]	R50-FPN	200	34.3	40.3	36.4	-	-	-	-
DetCo [26]	R50-FPN	200	34.7	40.1	36.4	-	-	-	-
BYOL [14]	R50-FPN	1000	41.4	40.6	36.7	24.5	23.2	13.0	12.5
SwAV [3]	R50-FPN	800	42.4	40.1	36.4	24.3	23.1	12.9	12.7
SoCo [25]	R50-FPN-Head	100	43.3	41.8	37.3	28.2	25.5	15.0	13.9
ADePT (separate)	R50-FPN-RPN-Head	100	43.2	**41.9**	**37.6**	**28.6**	**25.9**	**15.3**	14.1
ADePT (joint)	R50-FPN-RPN-Head	100	**43.6**	41.8	37.5	**28.6**	25.8	**15.3**	**14.2**

Table 1: Pre-trained ResNet50 models evaluated on object detection (AP_{bb} / $AP50_{bb}$) and instance segmentation (AP_{mk}) tasks using MS COCO and SODA10M. All methods are pre-trained on ImageNet. For the upper table, we directly quote performance reported in the cited articles. A Mask R-CNN detector is used in each case. The lower table reports our experimental results, where the detector is Mask R-CNN for MS COCO and Cascade R-CNN for SODA10M, respectively.

learned representations is then evaluated using the MS COCO `val2017` image subset. We report the metrics AP_{bb} for object detection and AP_{mk} for instance segmentation.

Results: The performance is reported in Table 1. We observe that our separately trained model consistently offers moderate improvements upon SoCo and other baselines in both object detection and instance segmentation metrics on MS COCO. We attribute improvements in both tasks to predominantly be due to our reduced localization error (see Appendix) and model sensitivities to related concepts. The joint training strategy provides overall competitive performance with SoCo on MS COCO, but achieves state-of-the-art performance on the challenging SODA10M benchmark. Generally, with smaller fine-tuning datasets (MS COCO), the performance gap between ADePT and SoCo becomes larger. In comparison with the performance of SoCo, this partially answers our **Q1** and we conclude that RPN pre-training provides a practical solution towards improving SSL performance. We provide further insight towards **Q1** in an error analysis ablation study in the Appendix. Our results also resolve **Q2**: the alignment between the pretext task and downstream task can bring improvement to overall performance.

Table 1 results might allow one to argue that absolute performance gains, observed when pre-training the RPN, are small. However MS COCO and SODA10M are large datasets (*e.g.* 1% of MS COCO contains $\sim 1.18K$ images). We thus conjecture that fine-tuning all model weights, in the investigated settings, may benefit SoCo and other SSL baselines disproportionately. With large-scale labeled data available, the RPN component (randomly initialized for SoCo and other baselines) can be deemed easier to fine-tune in comparison with both feature extractor and detector heads, due to the relatively small number of learnable parameters. To explore this point further, we design a few-shot detection experiment (Appendix), to better understand the contribution of RPN pre-training. The results from it show strong benefit from such pre-training. Together our results answer **Q3**: pre-training all architectural components improves label efficiency.

4 Conclusion

In this work we present ADePT, a self-supervised learning strategy that pre-trains the region proposal network (RPN) in a multi-stage detector. The alignment of both architecture and learning objectives result in improved downstream task performance. Through empirical study, we showed that (**A1**) RPN pre-training can reduce localization error, (**A2**) the alignment between the pretext and downstream tasks improve the overall performance, and (**A3**) RPN pre-training affords benefit to label-efficient scenarios.

References

[1] D. Bolya, S. Foley, J. Hays, and J. Hoffman. TIDE: A general toolbox for identifying object detection errors. In *ECCV*, pages 558–573, 2020.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NIPS*, volume 33, pages 9912–9924, 2020.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[5] K. Chen, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *ICCV*, pages 7546–7554, 2021.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[8] N. Dong, M. Maggioni, Y. Yang, E. Pérez-Pellitero, A. Leonardis, and S. McDonagh. Residual contrastive learning for image reconstruction: Learning transferable representations from noisy images. In *IJCAI*, pages 2930–2936, 2022.

[9] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *CVPR*, pages 5414–5423, 2021.

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[11] Z. Fan, Y. Ma, Z. Li, and J. Sun. Generalized few-shot object detection without forgetting. In *CVPR*, pages 4527–4536, 2021.

[12] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[14] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NIPS*, volume 33, pages 21271–21284, 2020.

[15] J. Han, X. Liang, H. Xu, K. Chen, H. Lanqing, J. Mao, C. Ye, W. Zhang, Z. Li, X. Liang, et al. Soda10m: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving. In *NIPS Track on Datasets and Benchmarks*, 2021.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, volume 28, pages 91–99, 2015.

[20] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 4080–4090, 2017.

[21] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[22] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3637–3645, 2016.

[23] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu. Frustratingly simple few-shot object detection. In *ICML*, pages 9919–9928. PMLR, 2020.

[24] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021.

[25] F. Wei, Y. Gao, Z. Wu, H. Hu, and S. Lin. Aligning pretraining for detection via object-level contrastive learning. In *NIPS*, volume 34, pages 22682–22694, 2021.

[26] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021.

[27] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021.

A Algorithmic Details

A.1 Architecture

The convolutional backbone of a multi-stage R-CNN architecture [13] firstly generates representations of an input image. Regions of interest (RoI) are then derived from these representations using a region proposal network (RPN). Regions identify where objects may be located and the model subsequently employs a detection head (*i.e.* RoI) in order to classify the same, shared, representations.

A.2 Selective Search

Prior to self-supervised pre-training, we generate unsupervised object proposals using the heuristic selective search algorithm [21]. The algorithm relies on colour and texture similarities to segment images of natural scenes into regions, which are subsequently merged to form the basis of the output bounding boxes. Default parameters are used in this process, with minimum segment size defined by $\texttt{min\_size} = 10$, output segment size and number by $\texttt{scale} = 500$, and the Gaussian kernel size for smoothing by $\sigma = 0.9$. Each bounding box $b = \{x,y,w,h\}$ is represented by its centre coordinate (x,y) and its width and height (w,h). We keep only boxes that satisfy $\frac{1}{3} \leq \frac{w}{h} \leq 3$ and $0.3 \leq \frac{\sqrt{wh}}{\sqrt{WH}} \leq 0.8$, where W and H are the height and width of the image.

A.3 Detector Head

We train a Fast R-CNN detector head architecture [12], using a self-supervised similarity loss. Recall our main manuscript defines a feature extractor network f_θ and we then define the Fast R-CNN layers as g_θ. The standard output layers of the detector are replaced by network heads pertaining to a projector p_θ and a predictor, q_θ. An "offline" encoder of the same architecture (*i.e.* defined as the *target* network in MoCo [17]) is maintained with parameters ξ. The analogous modules; f_ξ, g_ξ, p_ξ are updated as an exponential moving average of the *online* network parameters θ. Note that the momentum encoder does not require a final predictor head, nor does it include RPN layers.

View Augmentation During training, an input image x is augmented in order to define three different views; V_1,V_2,V_3. View V_1 is resized to 224×224, V_2 is a random crop of V_1 with a randomly selected scale in the range [0.5, 1.0]. View V_3 is a downsampled version of V_2, with resulting size 112×112. The bounding boxes associated with each view are similarly scaled and shifted according to the above transformations. For each view we also apply colour jitter and blurring as in BYOL [14]. The augmentations described enables learning of scale- and translation-invariances, in a similar fashion to SoCo [25].

Detection Loss Let $b = \{b_i\}_{i=1}^K$ be the K randomly chosen bounding boxes for an image as generated by selective search. Next, given views V_1, V_2, V_3 and each bounding box b_i, we extract the object-level representation from the backbone through the $\texttt{RoIAlign}$ method [16]

$$h_{i,\theta} = g_\theta(\texttt{RoIAlign}(f_\theta(V_1), b_i)), \tag{1}$$

$$h'_{i,\xi} = g_\xi(\texttt{RoIAlign}(f_\xi(V_2), b_i)), \tag{2}$$

$$h''_{i,\xi} = g_\xi(\texttt{RoIAlign}(f_\xi(V_3), b_i)). \tag{3}$$

The projector and predictor heads are then used to create the latent representations of the views

$$v_{i,\theta} = q_\theta(p_\theta(h_i)), \quad v'_{i,\xi} = p_\xi(h'_i), \quad v''_{i,\xi} = p_\xi(h''_i). \tag{4}$$

The full loss for the detector head contrasts the latent representations of the same proposal across different views by maximising their similarity.

$$\mathcal{L}_{\text{sim}}(v_\theta, v'_\xi, v''_\xi) = \\ \frac{1}{K}\sum_{i=1}^{K}\left(-\cdot\frac{2\langle v_{i,\theta}, v'_{i,\xi}\rangle}{\|v_{i,\theta}\|_2 \cdot \|v'_{i,\xi}\|_2} - \cdot\frac{2\langle v_{i,\theta}, v''_{i,\xi}\rangle}{\|v_{i,\theta}\|_2 \cdot \|v''_{i,\xi}\|_2}\right) \tag{5}$$

A symmetrical loss, that alternatively passes V_1 through the momentum encoder and $\{V_2,V_3\}$ through the online encoder is also produced, denoted as $\bar{\mathcal{L}}_{\text{sim}}(v_\xi, v'_\theta, v''_\theta)$. The total loss to train the detector head is then

$$\mathcal{L}_{\text{det}}(b, V_1, V_2, V_3) = \mathcal{L}_{\text{sim}}(v_\theta, v'_\xi, v''_\xi) + \bar{\mathcal{L}}_{\text{sim}}(v_\xi, v'_\theta, v''_\theta). \tag{6}$$

A.4 Region Proposal Network

The region proposal network (RPN) [19] component of two-stage object detectors consists of a fully convolutional network that simultaneously predicts object bounds and objectness scores at each spatial position. It works by placing a large number of *anchor boxes* on a spatial grid, from which it selects those most likely to contain an object, while adjusting their position and size to more closely match the object outline. The anchors are a set of fixed size / ratio rectangles, located at each point in the feature space. The RPN feeds the features extracted from the feature extractor (denoted as "backbone + FPN" in Fig. 2) through a set of convolutional layers to predict whether or not each such anchor corresponds to an object, in addition to anchor deltas that offset shape and position. By suppressing overlapping predictions and ranking the proposals by their "objectness" scores, the RPN can output a set of high quality proposals. More formally, let the feature extractor be f, the RPN convolutional layers, c, the objectness layer o, and anchor delta layer d. Then given an image x, the objectness scores are computed as $p = o \circ c \circ f(x)$ and the anchor deltas as $t = d \circ c \circ f(x)$. The loss to train the RPN is defined as

$$\begin{aligned} \mathcal{L}_{\text{RPN}}(p,t) = &\frac{1}{N_{\text{cls}}} \sum_{i=1}^{N_{\text{cls}}} \mathcal{L}_{\text{cls}}(p_i, p_i^*) \\ &+ \lambda \frac{1}{N_{\text{reg}}} \sum_{i=1}^{N_{\text{reg}}} p_i^* \mathcal{L}_{\text{reg}}(t_i, t_i^*), \end{aligned} \tag{7}$$

where p_i is the predicted objectness probability of anchor i and p_i^* is its ground-truth label. This label is 1 if either: it has highest IoU with a ground-truth box, or it has an IoU greater than 0.7 with any box. The objectness label is 0 if the IoU is less than 0.3, and any other anchors are ignored. The classification loss is a log loss over the binary objectness values. Additionally, t_i are the predicted anchor deltas that adjust the position and size of the anchor. The corresponding ground-truth deltas t_i^* are computed for the ground-truth box associated with the positive anchor i. The regression loss is a smooth $\mathcal{L}_1$ loss which only activates when $p_i^* = 1$ and is otherwise disabled. For further details, see [19].

A.5 Full Algorithm

Our whole detection architecture is trained in a self-supervised manner through two losses. The first is the bounding box regression loss that trains the RPN head towards high quality proposals, and the second is the similarity loss that drives the detection head towards discriminative features. Through ablations on the loss terms, we find that the backbone benefits from only receiving training signal from $\mathcal{L}_{\text{det}}$. By jointly training all architectural components, we arrive at a total loss of:

$$\begin{aligned} \mathcal{L}(p,t,b,V_1,V_2,V_3) = &\mathcal{L}_{\text{RPN}}(p,t) \\ &+ \mathcal{L}_{\text{det}}(b,V_1,V_2,V_3). \end{aligned} \tag{8}$$

During training, we perform stochastic gradient descent to minimize this loss.

B Training Details

B.1 Detector Head Pre-Training

We train the detector head with a scale-aware assignment strategy for assigning proposals to FPN levels, analogous to SoCo [25]. Typical supervised object detection training uses small batch sizes due to the large number of region proposals for each image, where K can be as high as 1000 [19]. In our setting we require a larger batch size to train the detector with the BYOL-style similarity loss [14]. We therefore require to adjust the number of proposals that power the detector head loss. In our experiments we set $K = 4$, following the ablation in SoCo. The low proposal count is mitigated in self-supervised setups through the use of multiple views and long training times.

B.2 RPN Pre-Training

The region proposal network (RPN) is trained by regressing to the bounding boxes produced by selective search. At each iteration of training, K random boxes are selected for each image. From that image, the feature pyramid network backbone generates a set of features $\{p2, p3, p4, p5\}$ that form

the input to the RPN. While it is common to also include the final pyramid features $p6$, the images used in our self-supervised pre-training, of size 224×224, are deemed too small for the resolution of $p6$ to be useful. The RPN generates a set of anchors at each level of the feature pyramid, with areas of $\{24^2,48^2,96^2,192^2\}$, respectively, and at three aspect ratios; $\{0.5,1.0,2.0\}$. Anchors are matched with target bounding boxes and two losses are used to update the network parameters, a classification log loss and a smooth $\mathcal{L}_1$ loss. When using RPN proposals in the detector head loss, we select the top 64 proposals before non-maximum suppression is applied, and finally the top K proposals are output.

In supervised object detection, ground-truth bounding boxes, as annotated by humans, are used to train the RPN using the loss described in Sec. A.4. Given our self-supervised setup, we rather mitigate a lack of ground-truth bounding boxes by alternatively considering pseudo ground-truth labels; we utilize bounding boxes generated via an unsupervised region proposal method. We here employ the selective search algorithm [21] for this purpose. This algorithm uses heuristic colour and texture similarity scores to create a set of bounding boxes for an input image. These bounding boxes are pre-computed before training and filtered such that only boxes that are large enough and within certain aspect ratio constraints are retained (further details can be found in the Appendix). During training, we randomly select K bounding boxes for each image in a batch, and treat them as ground-truth bounding boxes towards providing a valid learning signal to update the RPN.

B.3 Training Strategies

In the supervised detection literature, it is common to train RPN modules either *jointly*, in conjunction with other model architectural components, or *separately*, in a potentially alternating fashion [19]. We experiment with analogous strategies pertaining to both of these options for our alternative unsupervised setting.

We train ADePT under both the aforementioned "separate" and "joint" training strategies. We train each model for 100 epochs using $8\times$ NVIDIA V100 GPUs. Training takes ~3 days and ~4.5 days on ImageNet [7] for each strategy, respectively.

Separate Training We pre-train only the RPN component of the network. The feature extractor (backbone + FPN) and detector head are initialized from a SoCo baseline, pre-trained for 100 epochs. We use the LARS optimiser with a learning rate of 0.1 annealed to 0 with a cosine scheduler for 100 epochs.

Joint Training The joint training setup uses both the RPN loss as described in Eq. 7 and the detector loss (defined in Appendix B) that pre-trains the feature extractor and detector head. This strategy enables self-supervised training of the entire multi-stage architecture including: feature extractor, detector head and RPN from random initialization. We use the LARS optimiser with a learning rate of 0.1 annealed to 0 with a cosine scheduler for 100 epochs. The RPN model layers are updated using the $\mathcal{L}_{\text{RPN}}$ loss term (Eq. 7), exclusively.

ADePT Setting		COCO		COCO 10%		COCO 1%		SODA10M
Loss	Proposals	AP_{bb}	AP_{mk}	AP_{bb}	AP_{mk}	AP_{bb}	AP_{mk}	AP_{bb}
$\mathcal{L}_{\text{Det}}+\mathcal{L}_{\text{RPN}}$	SS + RPN	6.784	6.528	0.003	0.002	0.001	0.000	36.853
$\mathcal{L}_{\text{Det}}$	SS + RPN	0.564	0.771	0.020	0.018	0.015	0.019	0.006
$\mathcal{L}_{\text{Det}}+\mathcal{L}_{\text{RPN}}$	SS	41.158	37.002	27.783	25.073	14.508	13.332	43.168
$\mathcal{L}_{\text{Det}}$	SS	**41.677**	**37.342**	**28.576**	**25.794**	**15.161**	**13.938**	**43.570**

Table 2: Ablations on ADePT joint training show that the backbone benefits slightly from only receiving the detector head loss signal and that the RPN proposals tend to destabilize detector head training. SS: selective search.

B.4 Analysis of Joint Training

In contrast to separate training, we note that joint training can be affected by several confounding factors. Namely ADePT consists of two component losses; firstly the RPN pre-training loss, $\mathcal{L}_{\text{RPN}}$ (Eq. 1 in the main paper) and secondly the self-supervised loss that pre-trains the feature extractor and detector head (denoted as $\mathcal{L}_{\text{Det}}$ - see Appendix B for further details). During joint-training, the feature extractor is affected by both of these losses, while the detector head may receive proposals from either: solely selective search or selective search in conjunction with the RPN. In Table 2, we investigate how to best combine the component losses and the proposal-generation options.

We find that, compared with the quality of proposals, the loss signal to the feature extractor has limited effect on the performance, while a lone $\mathcal{L}_{\text{Det}}$ offers improved performance over $\mathcal{L}_{\text{Det}}+\mathcal{L}_{\text{RPN}}$. Additionally, as the RPN module is learning, by inspection we find that intermediate proposals are often noisy, which introduces a destabilising effect to the detector head loss. This causes the models trained with RPN proposals to fail on downstream tasks, leading us to consider only selective search proposals in $\mathcal{L}_{\text{Det}}$. Our joint training ADePT model corresponds to the final row of Table 2.

C Further Experiments

	Pre-training		Split 1					Split 2					Split 3				
	Method	Architecture	K=1	K=2	K=3	K=5	K=10	K=1	K=2	K=3	K=5	K=10	K=1	K=2	K=3	K=5	K=10
Novel	SoCo [25]	R50-FPN-Head	0.56	1.00	0.94	1.91	2.96	0.19	0.73	1.26	2.29	1.68	0.12	1.52	1.22	1.48	3.42
	ADePT (separate)	R50-FPN-RPN-Head	1.60	**2.32**	**1.64**	1.17	**3.52**	1.69	**1.94**	2.07	1.85	3.74	**3.03**	**2.13**	1.42	**3.56**	3.56
	ADePT (joint)	R50-FPN-RPN-Head	**2.14**	1.23	1.24	**1.93**	2.91	**2.53**	0.59	**2.80**	**5.55**	**4.18**	1.07	1.47	**1.43**	2.05	**5.72**
All	SoCo [25]	R50-FPN-Head	1.14	0.99	1.26	1.50	2.89	0.68	1.53	1.12	1.49	3.23	0.96	1.23	2.08	1.83	4.45
	ADePT (separate)	R50-FPN-RPN-Head	**1.72**	**1.35**	**1.63**	2.25	**4.53**	1.61	**1.87**	1.91	4.42	3.70	**1.60**	**1.91**	**2.51**	**2.42**	4.45
	ADePT (joint)	R50-FPN-RPN-Head	1.62	1.20	1.41	**2.59**	4.20	2.04	1.60	**2.35**	**4.52**	**4.31**	0.68	1.07	1.39	1.68	**4.60**

Table 3: Few-shot object detection performance (mAP50) on the PASCAL VOC dataset. Following the same protocol as [11], we evaluate the few-shot fine-tuned model weights on three different splits of novel classes (top) and novel + base classes (bottom). The detector is a Faster R-CNN and the model weights are pre-trained using ImageNet in a self-supervised fashion. Our approach consistently outperforms the SoCo baseline by a large margin. K denotes K-shot detection task.

C.1 Few-shot Evaluation

Experimental Setup: In addition to the transfer evaluation protocol described previously in Sec. 3, we further realize a new protocol to evaluate the performance of self-supervised learning under scenarios that exhibit extreme label-scarcity. The proposed protocol is inspired by the benchmark few-shot object detection evaluation convention adopted in [23, 11]. Few-shot object detection tasks consist of a set of base classes and a set of novel classes, where base classes have abundant labels and yet novel classes are provided with only few labels. Without loss of generality, we take [23] as an example. There are two training stages in [23], namely, a base training stage and a few-shot fine-tuning stage. During base training, standard supervised training is performed, and in the few-shot fine-tuning stage, the model is expected to learn to detect the novel classes given only K labelled instances for each class of interest, *i.e.* a K-shot learning task. In this work, we alternatively abandon the base training stage and pre-train the model (Faster R-CNN with a ResNet50 backbone + FPN) on ImageNet [7], *i.e.* no labelled data are involved in the first stage. Then, following [23, 11], we fine-tune the model via K-shot training [22, 20]. Finally we use a proxy evaluation task to assess both representation and fast adaptation capabilities of the pre-trained model weights. We note that such proxy evaluation protocols are commonly adopted in representation learning, *e.g.* [17, 6, 8], where only the last layer is fine-tuned with other layers fixed. As SoCo remains *unable* to pre-train an RPN component, towards fair comparison, we instead opt to fine-tune their entire model.

We perform the proposed few-shot evaluation on the **PASCAL VOC** dataset [10]. We follow the fine-tuning and evaluation setup in [23]. We use the `trainval07+12` set with the same data splits as [23] and report object detection performance on the `test07` set, using the standard VOC metric; *i.e.* mAP50.

Results: In Table 3 we report our few-shot learning results. We observe that ADePT can consistently outperform SoCo in the considered $K = \{1,2,3,5,10\}$ settings. Note, the unsupervised pre-training is performed on ImageNet, *i.e.* VOC data are only seen in the few-shot fine-tuning stage. This proxy evaluation evidences that ADePT has the ability to learn more useful representations than SoCo and shows better potential for fast-adaptation tasks, with few labels. With small K, the self-supervised pre-training in effect plays a critical role in the few-shot learning through the provision of strong initialisation. In comparison with Table 1, we highlight that the performance gain with RPN pre-training becomes yet larger. Furthermore, in Table 3, the performance gain of RPN pre-training is larger, specifically, for the novel classes *c.f.* the base classes, where only base classes are seen during the fine-tuning stage. This suggests that RPN pre-training aids the learning of representations that can quickly adapt to new tasks. Finally, we can provide some initial evidence towards answering **Q3**: RPN pre-training, with architectural alignment, is capable of bringing benefit to label-efficient regimes.

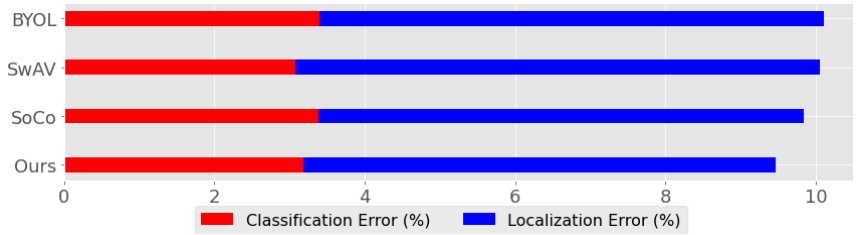


Figure 3: Localization errors contribute significantly to overall detection error rates. The effect can be observed by evaluating recent SSL approaches BYOL [14], SwAV [3], SoCo [25], here using the MS COCO dataset [18]. Models are pre-trained on ImageNet [7] and fine-tuned on MS COCO using Mask R-CNN [16]. Our ADePT method reduces the dominant localization error term. Full results are found in Table 4.

Based on the results in Table 1 and Table 3, we note that, although both of our training strategies can show improvement over SoCo, performance between our *separate* and *joint* training models is largely comparable in the majority of the investigated scenarios. The former strategy requires lower computation-cost, which may lead to it being favoured in practical situations.

C.2 Ablation Studies

C.2.1 Stratification of Error Types

Object detection is a complex task with a multitude of potential error types. Such errors can be categorised and quantified by tools such as TIDE [1]. In Table 4, we stratify these errors for the detection models fine-tuned on the full MS COCO dataset. The error types, from left to right are; classification (*Cls*), localisation (*Loc*), duplicates (*Dupe*), background objects (*Bkg*), missed objects (*Miss*), false positives (*FalsePos*), and false negatives (*FalseNeg*).

Method	Cls	Loc	Dupe	Bkg	Miss	FalsePos	FalseNeg
SoCo	3.39	6.45	**0.21**	**4.15**	6.89	**16.81**	14.57
ADePT (sep)	**3.18**	**6.28**	0.22	4.35	6.61	17.40	**13.59**
ADePT (joint)	3.31	6.44	0.22	4.40	**6.54**	17.45	13.75

Table 4: Stratified errors as computed by the TIDE toolbox [1]. ADePT (separate training) exhibits the lowest localization error, which is visualized in Fig 3. Other error types (*e.g.* background and false positives) are more challenging for our method. We conjecture that reliance on selective search proposals may cause the detector to "hallucinate" objects.

We observe that ADePT achieves the lowest localization error, with ADePT (separate) achieving a significant reduction with respect to the baselines. The classification errors of these models are additionally reduced compared to SoCo. Notably, ADePT has low error in the "missed objects" category along with low "false negative" error, which may be interpreted intuitively in terms of the rate of successfully locating *all* objects in images. Conversely, the "background object" error and "false positive" rates are higher, which we conjecture to illustrate that the greater reliance on selective search proposals produces a model that is prone to false positives, finding objects where they do not exist. We conclude that RPN pre-training can reduce the localization error, which corroborates our affirmative answer to **Q1**.

C.2.2 Qualitative Examples

In Fig. 1 and Fig. 4, we present qualitative examples of model predictions on the SODA10M and MS COCO datasets. These examples confirm what is shown by our error analysis: ADePT tends to find most objects in an image including small objects instances that benefit from RPN pre-training.

D Additional Visualizations

In Fig. 5 - Fig. 7, we provide additional visualizations to better understand the contributions and limitations of the proposed method. By comparing Fig. 5 and Fig. 6, we conjecture that RPN pre-training helps

(a) BYOL (b) SwAV (c) SoCo (d) ADePT (separate) (e) ADePT (joint)

Figure 4: Qualitative comparison on the MS COCO dataset [18]. Top: our models have the highest confidence in their predictions and ADePT (joint) additionally detects the person in the distance despite difficult illumination. Bottom: all models successfully capture the large stop-sign in the foreground. Notably, our models detect the "person" object in a small "picture". Both examples imply that RPN pre-training can improve the localization of small objects.

more under normal imaging conditions when the pre-training and fine-tuning datasets have similar distribution. As shown in Fig. 7, ADePT fails under several challenging imaging conditions, such as darkness and overexposure, when the pre-training and fine-tuning datasets are drastically different. In practice, we suggest that the pre-training dataset should be selected to be similar to the fine-tuning dataset.

(a) BYOL (b) SwAV (c) SoCo (d) ADePT (separate) (e) ADePT (joint)

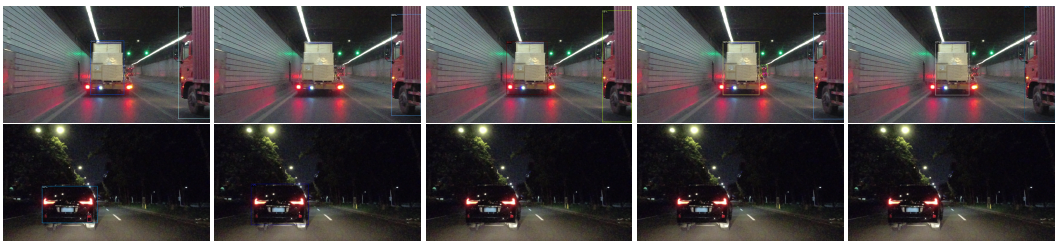
Figure 5: Example predictions on the COCO dataset [18]. BYOL [14] fails to catch the sheep on the left-hand side of the picture. SwAV [3] overcounts sheep by incorrectly predicting a (single) sheep in the middle of the picture as multiple instances. SoCo [25] fails to detect a sheep in the middle of the picture and also makes a prediction that incorrectly splits a single sheep into two (right-hand side). Our ADePT models (both separate and joint training) catch **all** instances with high confidence. **Best viewed with digital zoom**.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1.

(a) BYOL (b) SwAV (c) SoCo (d) ADePT (separate) (e) ADePT (joint)

Figure 6: An example failure case of ADePT predictions on the COCO dataset [18]. All methods successfully detect the person in the middle of the image, while missing additional smaller objects. BYOL and SwAV manage to detect the TV in the background under blurry imaging conditions. SoCo can detect the partially occluded phone in the hand of the person. Our models fail to detect the TV and the phone objects and the joint training also introduces a false positive, due to the background appearance, under this challenging situation.



(a) BYOL (b) SwAV (c) SoCo (d) ADePT (separate) (e) ADePT (joint)

Figure 7: Example cases of prediction failure, pertaining to the SODA10M dataset [15]. Top: BYOL and SwAV both fail to detect a truck. SoCo successfully detects three trucks. ADePT (separate) incorrectly predicts the small, distant truck as two separate vehicles while ADePT (joint) fails to detect it. Bottom: BYOL and SwAV can detect the dark car in the challenging night scenario, while SoCo and ADePT both fail to predict it. **Best viewed with digital zoom**.