Data-Efficient Image Recognition with Contrastive Predictive Coding

Olivier J. Hénaff, Ali Razavi, Carl Doersch,
S. M. Ali Eslami[*], Aaron van den Oord[*]
DeepMind

Abstract

Large scale deep learning excels when labeled images are abundant, yet data-efficient learning remains a long-standing challenge. While biological vision is thought to leverage vast amounts of unlabeled data to solve classification problems with limited supervision, computer vision has so far not succeeded in this 'semi-supervised' regime. Our work tackles this challenge with Contrastive Predictive Coding, an unsupervised objective which extracts stable structure from still images. The result is a representation which, equipped with a simple linear classifier, separates ImageNet categories better than all competing methods, and surpasses the performance of a fully-supervised AlexNet model. When given a small number of labeled images (as few as 13 per class), this representation retains a strong classification performance, outperforming state-of-the-art semi-supervised methods by 10% Top-5 accuracy and supervised methods by 20%. Finally, we find our unsupervised representation to serve as a useful substrate for image detection on the PASCAL-VOC 2007 dataset, approaching the performance of representations trained with a fully annotated ImageNet dataset. We expect these results to open the door to pipelines that use scalable unsupervised representations as a drop-in replacement for supervised ones for real-world vision tasks where labels are scarce.

1. Introduction

Current successes in deep learning have almost universally relied on large annotated datasets. However, for many interesting applications labeling is prohibitively expensive, and these applications would benefit from methods that perform well with only small amounts of labeled data. For instance, in medical imagery, obtaining annotated data from medical experts is time consuming and expensive, limiting its availability. 3D annotations such as surface normals or 3D poses cannot easily be labeled with a point-
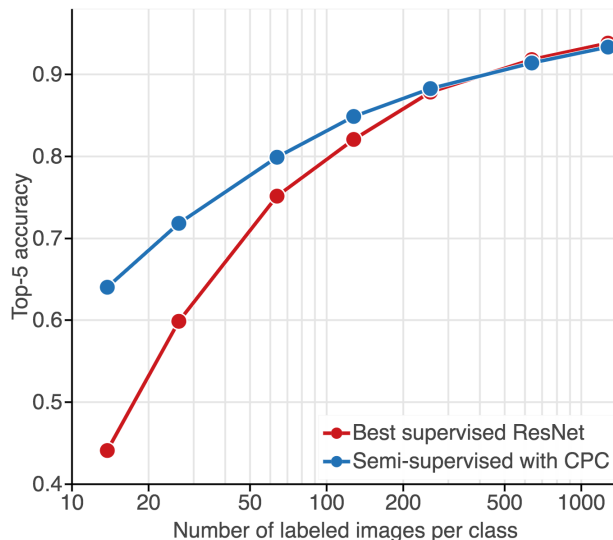
---
[*] Equal contribution

Figure 1. Classification accuracy as a function of the number of labeled examples. The performance of supervised methods (red curve) degrades substantially as the amount of labeled data decreases. Regularizing these methods with large amounts of unlabeled examples (blue curve) greatly alleviates this degradation.

and-click interface. And in more general domains involving autonomous agents, such as vehicles and robots deployed in the real world, it is not even clear what should be labeled in order to drive good behavior. Failures can often result in catastrophic damage to the agent or the environment.

Nevertheless, in many of these problem settings, unlabeled data is often readily accessible and available in large quantities. The idea of *semi-supervised learning* is to leverage unlabeled data to improve task performance when few labeled examples are available [10, 70]. One class of semi-supervised techniques consists in propagating the knowledge learned from the subset of labeled examples to unlabeled ones [23, 39], and learning from this new information. Another class of semi-supervised techniques propose to directly learn a representation from unlabeled data using mechanisms which can broadly be grouped into two categories: *generative models* that learn compact repre-

sentations that are capable of predicting pixel-level details [34, 54, 12], and *self-supervised methods* that formulate alternative prediction tasks in which missing information is imputed from other parts of the image [33]. Although many of these self-supervised objectives formulate prediction in terms of missing pixel-level details such as context or color [14, 69, 46, 38, 64], another class of methods defines prediction in higher-level, learned representation spaces [25, 16, 47, 49, 51]. These latter techniques are appealing because they may encourage the learning of abstract properties and semantics of images while ignoring the low-level details of image pixels. Regardless, representations learned from unlabeled images can then used (and often fine-tuned) for the task at hand, which, taken together, constitutes the final semi-supervised technique.

A recent approach for representation learning that has demonstrated strong empirical performance in a variety of modalities is Contrastive Predictive Coding (CPC, [49]). CPC encourages representations that are stable over space by attempting to predict the representation of one part of an image from those of other parts of the image. Although CPC training is completely unsupervised, its learned features tend to linearly separate image classes, as evidenced by state-of-the-art accuracy for ImageNet classification with a simple linear network [49]. While this result suggests that CPC can facilitate image classification, the aforementioned evaluation still uses a large number of labeled examples to train its classifier, limiting its practical applicability. Furthermore, the final performance of this linear classifier is still far from that of fully supervised networks trained on ImageNet.

In this work, we show that a relatively straightforward approach based on our proposed improvements to the CPC model can alleviate both of these problems. First, we show that, with architectural optimizations, CPC feature encoders can be scaled to much larger networks, which can therefore absorb more useful information from unlabeled data, resulting in features that separate image categories better. In particular, we show that a linear classifier trained on top of these representations outperforms even a fully supervised AlexNet network [37] in terms of both Top-1 and Top-5 accuracy on ImageNet.

Second, we explore the use of this representation for classification with a small number of labels, as few as 1% of the entire ImageNet dataset. In this regime, our semi-supervised method results in an 20% absolute improvement in Top-5 accuracy over state-of-the-art supervised methods and a 10% absolute improvement over state-of-the-art semi-supervised methods. Unlike previous semi-supervised results, performance remains strong as more labeled examples are used, and even matches fully-supervised performance when the full ImageNet training set is used, suggesting that the learned features may also lend themselves to

online learning settings.

Third, we investigate the applicability of this representation for transfer learning. Using our unsupervised representation as a feature extractor for image detection on the PASCAL 2007 dataset leads to state-of-the-art performance compared to other self-supervised transfer methods. Importantly, this result approaches that found for supervised transfer learning.

Finally, we explore different methods for semi-supervised learning, and find that the standard approach— end-to-end fine-tuning—is not necessarily optimal. In fact, we find that CPC features can be used without any retraining by instead training a deep network on top of the frozen features. This approach yields almost the same performance as fine-tuning, yet with significantly reduced computational cost. This result is interesting because it echoes results from natural language processing, where unsupervised features such as word2vec [42] and BERT [13] provide strong performance across many tasks without retraining, which simplifies training pipelines and reduces computational requirements.

2. Related Work

The promise of unsupervised representation learning has existed since the earliest days of learning in computer vision [20], inspired by the fact that humans and animals need little supervision to perform a range of complex visual tasks. The intuition that a good representation should faithfully reconstruct the input data was soon exploited to achieve representations that resemble simple cells in the brain [48]. The idea was later formalized in the context of probabilistic generative models [27], which have become a staple of modern unsupervised learning [28, 22, 35, 56].

A key problem that generative models must overcome, however, is that not all low-level details are equally important in visual perception: for instance, lighting, compression artifacts, and the exact position of texture elements are irrelevant to most downstream tasks. This resulted in some early attempts to structure representations with losses that did not rely on reconstruction: notably, slow feature analysis [19, 63] used the persistence of objects in videos to learn representations that generalized across views.

Recent years have witnessed a proliferation of such unsupervised objectives. For example, simply asking a network to recognize the spatial layout of an image, without reconstructing it, led to representations that transferred to popular vision tasks such as classification and detection [14, 46, 49]. Other works showed that color [69, 38], image orientation [21], and data augmentation [17] can support useful self-supervised tasks from single images.

Extra information that occurs with images can allow for even richer objectives to train representations. Video is a particularly strong source of cues, where notable works

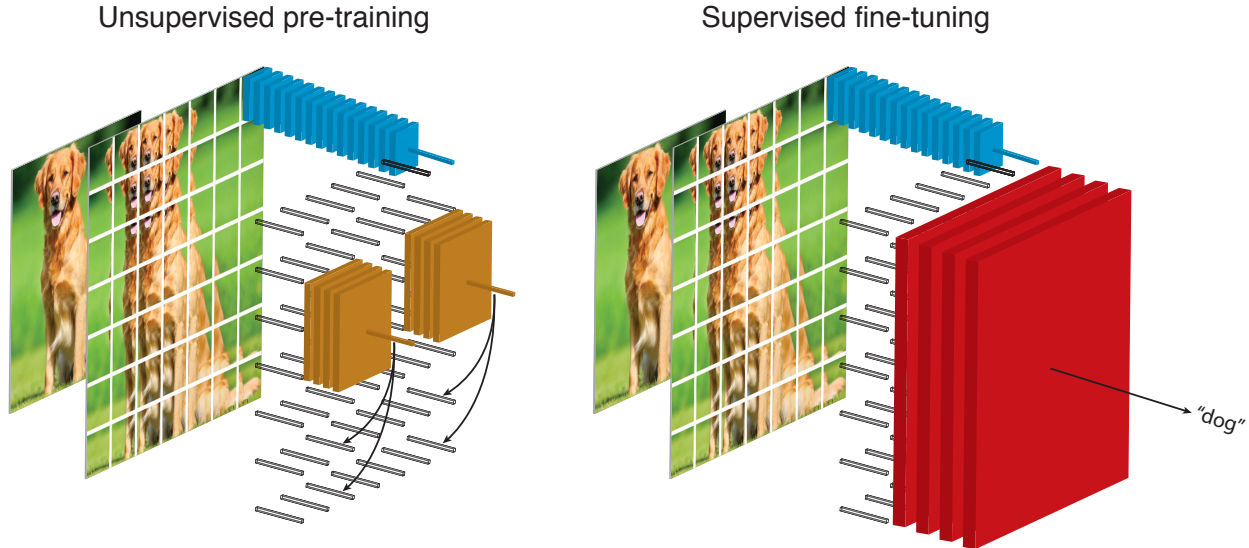Unsupervised pre-training   Supervised fine-tuning

Figure 2. Overview of the framework for semi-supervised learning with Contrastive Predictive Coding. Left: unsupervised pre-training with a spatial prediction task. First, an image is divided into a grid of overlapping patches. Each patch is encoded independently from the rest with a feature extractor (blue) which terminates with a mean-pooling operation, yielding a single feature vector for that patch. Doing so for all patches yields a field of such feature vectors (wireframe vectors). Feature vectors above a certain level (in this case, the center of the image) are then aggregated with a context network (brown), yielding a row of context vectors which are used to linearly predict (unseen) features vectors below. Right: using the CPC representation for a classification task. Having trained the encoder network, the context network is discarded and replaced by a classifier network (red) which can be trained in a supervised manner. For some experiments, we also fine-tune the encoder network (blue) for the classification task.

have leveraged object tracking [61], frame ordering [43], and object boundary cues [40, 50]. Beyond video, information about camera motion [1, 32], scene geometry [67], or sound [3, 4] can all serve as natural sources of supervision. One particularly important domain for self-supervised learning is robotics, where representations can be learned by interacting with the environment [2, 53, 52, 60], and where success greatly depends on making effective use of unlabeled data. Tasks can often be combined to form even more powerful learning signals, allowing larger networks to be trained successfully [15, 66, 36].

While many of these tasks require predicting fixed, low-level aspects of the data, another class of *contrastive* methods formulate their objectives in learned representation spaces. To avoid the collapse of their representations to a trivial solution, these objective are formulated as classification problems, in which the 'predicted' representation must be recognized amongst a set of distracting 'negative' examples. For example, camera motion [1, 32], tracking [61], text proximity [41], and other cues [11, 62, 30, 60, 29] can be used to find examples of features that should be 'similar' or predictable from one another, whereas negative examples are usually sampled randomly.

Other approaches to semi-supervised learning train simultaneously on labeled and unlabeled data. Many such works have roots in early work on bootstrapping [57] and co-training [6] from text (see [71] for a survey). An increasingly popular approach in deep learning consists in propagating information learned from labeled images to unlabeled ones [23, 39] while enforcing invariance to perturbations [44, 59, 65]. In relying on labeled data to provide a learning signal for unlabeled data, these methods could be limited in settings with very low labeled data.

3. Contrastive Predictive Coding

Contrastive Predictive Coding (CPC, [49]) is a self-supervised objective that learns from sequential data by predicting the representations of future observations from those of past ones. When applied to images, CPC operates by predicting the representations of patches below a certain level from those above it (Figure 2). These predictions are evaluated using a contrastive loss, in which the network must correctly classify the 'future' representation amongst a set of unrelated 'negative' representations. This avoids trivial solutions such as representing all patches with a constant vector, as would be the case with a mean squared error loss. In the following, we describe each of these steps in detail.

3

3.1. Prediction Task

Every input image is first divided into a set of overlapping patches $x_{i,j}$, each of which is encoded with a deep residual network [26] and spatially mean-pooled into a single vector. Specifically, we divide a 256×256-pixel image into 64×64 patches with a 32-pixel overlap, resulting in a 7×7 grid of feature vectors $z_{i,j} = f_\theta(x_{i,j})$.

To make predictions, a deep, masked convolutional network is then applied on the 7×7 grid of feature vectors. The network is masked such that the receptive field of a given output neuron can only see inputs that lie above it in the image. This *context* network is fully convolutional such that the output is also a 7×7 grid of context vectors $c_{i,j} = g_{\text{context}}(z_{i,j})$.

The prediction task then consists in predicting 'future' feature vectors $z_{i+k,j}$ from current context vectors $c_{i,j}$, where $k > 0$. The predictions are made linearly: given a context vector $c_{i,j}$, a prediction length $k > 0$, and a prediction matrix W_k, the predicted feature vector is

$$\hat{z}_{i+k,j} = W_k c_{i,j}$$

3.2. Contrastive Loss

The quality of this prediction is then evaluated using a contrastive loss. Specifically, the goal is to correctly recognize the target among a set of randomly sampled patch representations $\{z_l\}$ from the dataset. We compute the probability assigned to the target using a softmax, and evaluate this probability using the usual cross-entropy loss. Summing this loss over locations and prediction offsets, we arrive at the CPC objective:

$$\mathcal{L}_{\text{CPC}} = -\sum_{i,j,k} \log p(z_{i+k,j} | \hat{z}_{i+k,j}, \{z_l\})$$

$$= -\sum_{i,j,k} \log \frac{\exp(\hat{z}_{i+k,j}^T z_{i+k,j})}{\exp(\hat{z}_{i+k,j}^T z_{i+k,j}) + \sum_l \exp(\hat{z}_{i+k,j}^T z_l')}$$

The *negative samples* $\{z_l\}$ are taken from other patches in the image and other images in the mini-batch. This loss is called InfoNCE [49] as it is inspired by Noise-Contrastive Estimation [24, 45] and has been shown to maximize the mutual information between $c_{i,j}$ and $z_{i+k,j}$ [49].

4. Methods

4.1. Unsupervised learning with CPC

Following recent successes showing that increasing network capacity and the scale of training result in improved performance [15, 7], we seek to grow our architecture while also maximizing the supervisory signal we obtain from each image. We aim to make the training efficient while not allowing the network to fall into 'trivial shortcuts': ways of solving the problem without learning semantics.

Our first revision to the existing CPC algorithm is to make the network larger. While CPC originally used a ResNet-101-style [26] architecture to represent each patch, we developed a deeper and wider ResNet for this task. Specifically, the third residual stack of ResNet-101 contains 23 blocks, with a 1024-dimensional feature maps and 256-dimensional bottleneck layers. We increased the model's capacity by growing this component to 46 blocks with 4096-dimensional feature maps and 512-dimensional bottleneck layers, and call the resulting network ResNet-170.

However, extremely large architectures are more difficult to train efficiently. This is aggravated by the fact that CPC must be trained on patches: in order for the CPC objective to be a non-trivial task, the representation used to make the prediction for a given patch should not have access to that patch. This means that the input must be in the form of relatively small patches (i.e. 64 pixels), and the predicted patches must be a reasonably large distance away from the patches used to make the prediction. Early works on context prediction with patches used batch normalization [31, 14] to improve training speed. However, we find that for large architectures, batch normalization results in poor performance because it enables the model to find a trivial solution to the CPC objective. Indeed, high capacity networks can learn to communicate information between patches through the batch statistics of batch normalization layers. However we find that we can reclaim much of batch normalization's training efficiency using layer normalization [5] instead.

Given an effective, high-capacity architecture, we next develop a sufficiently challenging task to train it. We first double the supervisory signal for each image by predicting in the upward direction (i.e. patches spatially lower are aggregated to predict the representation of patches above) as well as the downward direction (CPC originally used only the downward direction). These two directions use different context networks. We also find that additional augmentation on the patches can result in substantial improvements. First, we adopt the 'color dropping' approach of [14], which randomly drops two of the three color channels in each patch. We also randomly flip patches horizontally. Furthermore, we spatially jitter individual patches by randomly cropping them to 56x56 patches and padding them back to their original size.

The upshot of this additional task complexity is that the CPC objective becomes very difficult, even for a model with such high capacity. In practice, we found that making the task harder increases rather than decreases performance in downstream tasks. Indeed if the network can learn to solve the task using low-level patterns (e.g. slowly varying color,

or straight lines continuing between patches), it need not learn any semantically meaningful content. By augmenting the low-level variability across patches, we remove these low level cues while also making the task harder, forcing the network to solve the task by extracting high-level structure.

4.2. Semi-supervised learning with CPC

We investigate two ways to integrate the CPC objective with a supervised classification task. The first *frozen* regime consists in optimizing a feature extractor f_θ solely for the CPC objective. Its parameters are then fixed and a classifier g_ϕ is optimized to discriminate the output of the feature extractor. More formally, given a dataset of N images $\{x_n\}$

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\text{CPC}}[f_\theta(x_n)].$$

And given a (potentially much smaller) dataset of M labeled images $\{x_m, y_m\}$

$$\phi^* = \arg\min_\phi \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{\text{Sup}}[g_\phi \circ f_{\theta^*}(x_m), y_m].$$

We also investigate a *fine-tuning* regime in which the feature extractor is allowed to accommodate the supervised objective. Specifically, we initialize the feature extractor and classifier with the solutions θ^*, ϕ^* found in the previous learning phase, and fine-tune the entire network for the supervised objective. To ensure that the feature extractor does not deviate too much from the solution dictated by the CPC objective, we use a smaller learning rate and early-stopping.

Whereas the CPC objective requires the feature extractor f_θ to be applied separately to overlapping patches, in the semi-supervised learning phase it can be applied directly to the entire image. This reduces overall computation by a factor of 3–4, thereby accelerating training and reducing the memory footprint. In order to mitigate the domain mismatch between unsupervised learning on patches and supervised fine-tuning on whole images, we use symmetric padding in all convolutions as well as the spatial jittering during the unsupervised pre-training.

When training the network for image classification, the classifier g_ϕ is an 11-block ResNet architecture with 4096-dimensional feature maps and 1024-dimensional bottleneck layers. The supervised loss $\mathcal{L}_{\text{Sup}}$ is the cross entropy between model predictions and image labels. When training the network for image detection, we use the Faster-RCNN architecture and loss, without any modification [55].

5. Results: ImageNet classification

First, in Section 5.1, we investigate our model's ability to linearly separate image classes, a standard benchmark in

Method	Top-1	Top-5
Motion Segmentation (MS) [50]	27.6	48.3
Exemplar (Ex) [17]	31.5	53.1
Relative Position (RP) [14]	36.2	59.2
Colorization (Col) [69]	39.6	62.5
Combination of		
MS + Ex + RP + Col [15]	-	69.3
CPC [49]	48.7	73.6
Rotation + RevNet [36]	55.4	-
CPC (ours)	**61.0**	**83.0**

Table 1. Comparison to linear separability of other self-supervised methods. In all cases a feature extractor is optimized in an unsupervised manner, and a linear classifier is trained using all labels in the ImageNet dataset.

unsupervised representation learning. In Section 5.2, we assess the performance of purely supervised networks on ImageNet with various amounts of labeled training data. Then, in Section 5.3 we compare the performance of our proposed semi-supervised method with that of the purely supervised baselines and prior art.

5.1. Linear separability

Following prior work in self-supervised learning, we re-evaluated the CPC objective in terms of its ability to linearly separate image classes if given all the training labels, which is an informative metric given the connection between linear separability and classification complexity [18]. As in [49], we trained a CPC feature extractor on the ILSVRC ImageNet competition dataset images [58]. To evaluate the representations we then trained a linear classifier on top of the spatially mean-pooled CPC features, using all the labels in the training set. Note that the mean pooling reduces the feature to a single spatial cell, so the dimensionality of the features we are linearly classifying is 4096. This is roughly half the dimensionality of the features used in competing methods, which makes the linear separation problem harder. Despite this, our improved CPC architecture improves upon previously published results by a significant margin (Table 1). Specifically, we increase the state of the art Top-1 accuracy from 55.4% [36] to 61.0%, a 12.3% absolute improvement over the previously published CPC method. With 61.0% Top-1 and 83.0% Top-5 accuracies, our self-supervised method is the first to surpass the performance of the AlexNet model (whose accuracies are 59.3% and 81.8%, respectively).
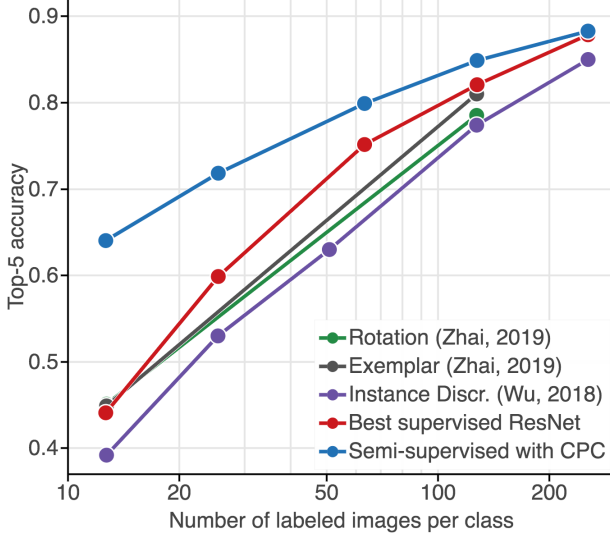
Figure 3. Comparison to other methods for semi-supervised learning via self-supervised learning followed by supervised fine-tuning. Blue: semi-supervised learning with CPC. Purple: semi-supervised learning with instance discrimination [64]. Green: semi-supervised learning with rotation prediction [68]. Grey: semi-supervised learning with exemplar learning [68]. Red: our supervised baseline.

Labeled data	1%	10%
Method	**Top-5 accuracy**	
Supervised baseline	44.10	82.08
Methods using label-propagation:		
Pseudolabeling [68]	51.56	82.41
VAT [68]	44.05	82.78
VAT + Entropy Minimization [68]	46.96	83.39
Unsup. Data Augmentation [65]	-	88.52
Rotation + VAT + Ent. Min. [68]	-	**91.23**
Methods only using representation learning:		
Instance Discrimination [64]	39.20	77.40
Exemplar [68]	44.90	81.01
Exemplar (joint training) [68]	47.02	83.72
Rotation [68]	45.11	78.53
Rotation (joint training) [68]	53.37	83.82
CPC (ours)	**64.03**	**84.88**

Table 2. Comparison to other methods for semi-supervised learning using 1% or 10% of labeled data. *Representation learning* methods learn a representation in an unsupervised manner and use it for classification. The classifier only considers labeled examples, and is only constrained by the supervised objective.

5.2. Low-data classification: fully-supervised

In order to investigate the efficiency of modern recognition architectures, we start by evaluating the performance of a well-tuned, purely supervised network as a function of the amount of labeled data it can train from. In only considering the labeled examples, these experiments serve as a baseline for semi-supervised experiments. We vary the proportion of labeled data from 1% to 100% logarithmically, and train a separate network on each subset. We then evaluate each model's performance on the publicly available ILSVRC validation set. We use data augmentation and extensively tune a number of hyperparameters (using a separate validation set), including model capacity, regularization strengths and optimization details for models trained on different subsets in order to maximise performance of the baseline.

As can be seen in Figure 1, with decreasing amounts of data, the model tends to overfit more severely. Although we increase the amount of regularization correspondingly, the model's performance drops from 93.83% accuracy (when trained on the entire dataset) to 44.10% accuracy (when trained on 1% of the dataset, see Figures 1, 3, red curve).

5.3. Low-data classification: semi-supervised

We next evaluate our semi-supervised method on this same task. Here, we pretrain our feature extractor on the entire unlabeled ImageNet dataset, and learn the classifier and fine-tune using a subset of labeled images. Figures 1 and

3 (blue curve) shows our results. In contrast to the purely supervised method, learning the majority of the network's parameters with CPC, for which we have large amounts of data, greatly alleviates the degradation in test accuracy as the amount of labeled data decreases. Specifically, our semi-supervised model's accuracy in the low-data regime (including only 1% of labeled images) is 64.03%—nearly 20% better than our supervised network—while retaining a high accuracy in the high-data regime (including all labels, Top-5 test accuracy = 93.35%).

We now compare our approach to other means of semi-supervised learning. These methods fall into two broad classes: those that use only representation learning and those that also use techniques such as label-propagation. The former class of methods, like ours, learn a representation in an unsupervised manner and then fine-tune it for classification. Instance discrimination [64], rotation prediction [21] and exemplar learning [17] are alternative methods for self-supervised learning that have been used for semi-supervised learning. Figure 3 shows that these methods do not substantially improve upon supervised baselines, making CPC the only self-supervised objective to surpass supervised learning in this regime. The authors from [68] investigate jointly training their model for both supervised and un-

6

supervised objectives (improving their best result by 8% in the 1% labeled-data regime, see table 2), thereby surpassing the supervised baseline. Our method represents a 10% improvement over theirs, but their results suggest that it could be further improved by jointly training the supervised and CPC objectives.

The second class of methods attempts to propagate the knowledge extracted from the subset of labeled examples to unlabeled examples while being invariant to augmentation or other perturbations[68, 65]. These methods, including Virtual Adversarial Training (VAT, [44]) and entropy minimization [23] are not sufficient by themselves to match our results, again falling short by 1.5% for 10% of labeled data and 17% for 1% of labeled data. When combined with the rotation prediction objective, these techniques improve its performance by 7.41% (reaching an impressive 91.23%) given 10% of labeled data. These results suggest that in this mid-data regime, our approach could also benefit from label propagation-based techniques. These methods have yet to be successfully applied in the low-data regime with only 1% of labeled data, suggesting that the unsupervised representation learning approach may be necessary in this more challenging regime.

6. Results: Transfer to PASCAL detection

Real-world applications often involve a dataset and task that are separate from the large (un)annotated dataset that is available for pre-training. A useful unsupervised learning objective is therefore one that trains a representation which transfers well to a novel dataset and task. To investigate whether this is the case for the representation learned with CPC, we evaluated its performance on image detection on the PASCAL dataset. For this, we again used the CPC representation trained on ImageNet, and placed a standard Faster-RCNN image detection architecture on top. As before, we first trained the Faster-RCNN model while keeping the CPC feature extractor fixed, then fine-tuned the entire model end-to-end. Table 3 displays our results compared to other methods. Most competing methods, which optimize a single unsupervised objective on ImageNet before fine-tuning on PASCAL detection, attain around 65% [17, 50, 69, 14, 8]. Leveraging larger unlabeled datasets increases their performance up to 67.8% [9]. Combining multiple forms of self-supervision enables them to reach 70.53% [15]. Our method, which learns only from ImageNet data using a single unsupervised objective, reaches 70.6% when equipped with a ResNet-101 feature extractor (as for most competing methods [15] but not all [8, 9]). Equipped with the more powerful ResNet-170 feature extractor, our method approach reaches 72.1%. Importantly, this result is only 2.6% short of the performance attained by purely supervised transfer learning, which we obtain by using all ImageNet labels before transferring to PASCAL.

Method	mAP
Transfer from labeled ImageNet:	
Supervised - ResNet-152	74.7
Transfer from unlabeled ImageNet:	
Exemplar (Ex) [17]	60.9
Motion Segmentation (MS) [50]	61.1
Colorization (Col) [69]	65.5
Relative Position (RP) [14]	66.8
Combination of	
Ex + MS + Col + RP [15]	70.5
Deep Cluster [8]	65.9
Deeper Cluster [9]	67.8
CPC - ResNet-101	70.6
CPC - ResNet-170	**72.1**

Table 3. Comparison of PASCAL 2007 image detection accuracy to other transfer methods. The first class of methods learn from unlabeled ImageNet data and fine-tune for PASCAL detection. The second class learns from the entire labeled ImageNet dataset before transferring. All results are reported in terms of mean average precision (mAP).

7. Analysis

What are the factors that contribute to the success of CPC in efficient image recognition? Our approach combines several elements: a particular architecture, unsupervised learning followed by fine-tuning. Which of these matter to its final performance, and in which data-regime? We start by evaluating the effect of the training scheme, then investigate that of model architecture and learning.

7.1. Training schemes

In order to dissect the contributions of unsupervised learning and fine-tuning, we evaluate our model accuracy at two stages: before and after fine-tuning. Across the range of labeled-data regimes, fine-tuning increases accuracy by approximately 1% (see Figure 4, light and dark blue curves). This absolute improvement is important in the high-data regime, where it is necessary to close the gap with supervised methods, but fairly marginal in the low-data regime (i.e. relative to the 20% improvement over supervised methods). Importantly, these results show that, even without fine-tuning the feature extractor, CPC excels at classification tasks in low-data regimes. Besides showing the generality of representations learned by CPC, this is an attractive feature from a practical standpoint. Using a fixed feature extractor, instead of one that needs fine-tuning, more than halves the amount of computation needed during su-
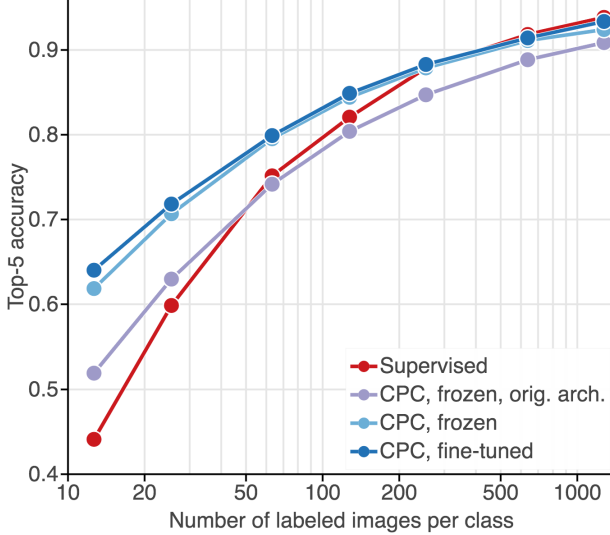
Figure 4. Contribution of unsupervised learning and fine-tuning to recognition performance. Light blue: classification performance of an frozen feature extractor followed by a supervised classifier. Purple: similarly, but with the original CPC architecture. Dark blue: classification performance of the fine-tuned model. Red: fully supervised baseline.
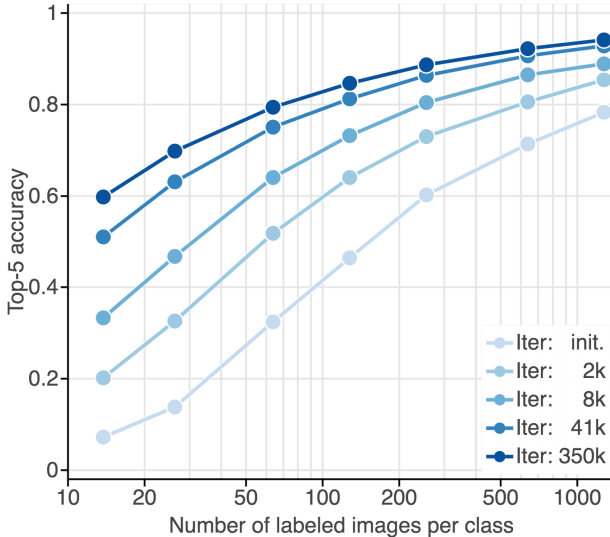

Figure 5. Image recognition accuracy over the course of CPC training. Without training, the ResNet-170 architecture achieves very low performance across data regimes. Over the course of training, this performance increases rapidly, reaching our final result after 350k iterations.

pervised training (since backward computations are more expensive than forward computations for CNNs) and can greatly reduce the memory footprint as only the final feature activations need to be retained. Training can be further accelerated by extracting and storing the features of a target

dataset offline, and only training a classifier on top of them.

We believe these results, combined with relative ease of use, motivate pretrained CPC features as a general and efficient off-the-shelf representation that should be considered for a variety of downstream tasks.

7.2. Architectures

Thus far, we have presented results using a particular ResNet architecture. To what extent do these results depend on the architecture we have chosen? How do they compare to the standard architecture employed in the original CPC work? We trained a ResNet classifier on top of CPC features learned with a smaller ResNet architecture as the feature encoder, identical to the one used in the original work (Figure 4, purple curve). This model performs significantly worse across the data-regimes we tested. In particular, it only achieves 52% accuracy in the low-data regime (1% of the labels), 12% less than our larger model. This is consistent with our linear classification results, in which our improved model increased Top-5 accuracy by 9%. This demonstrates that there is still much to be gained from increasing the scale of the architecture in self-supervised learning.

7.3. Learning dynamics

Given that our new large model architecture is important to obtain strong performance from CPC, we wish to verify that the CPC is still the main factor driving performance. Therefore we ask, how does the representation improve as a function of unsupervised training time? We investigated this question by re-training classifiers on top of CPC features at different moments over the course of training. Initially, we find that a randomly initialized ResNet with our larger architecture affords no benefit whatsoever: its accuracy in the low-data regime is below 10% (Figure 5, lightest blue). Over the course of CPC training, however, these results improve rapidly; 40,000 iterations are sufficient to surpass supervised methods in the low-data regime (Figure 5, darker blue), even though many of the parameters are not fine-tuned. After 350,000 iterations we approach the results presented previously (Figure 5, darkest blue), indicating that the CPC objective indeed plays a crucial role in our results.

8. Conclusions

Our results suggest that previous works were far from exhausting the potential of context as a supervisory signal for visual representation learning. By building a more powerful architecture to solve a CPC task, and increasing the difficulty of the CPC task, we trained a representation that outperforms all previous methods on ImageNet by a large margin, even when trained with as few as 13 images per

category. These features give almost equally strong performance without fine-tuning, suggesting the potential for generic, unsupervised features applicable to many vision tasks. However, while this paper only uses spatial predictions within an a single explores only context within a single image in order to simplify comparisons and thoroughly explore design choices, there are a multitude of other prediction tasks that may boost these results further, as suggested by [15]. An ideal task should incorporate time and other modalities, and we believe that contrastive feature prediction may serve as a unifying basis for many of these. Given the rapid improvement in self-supervised feature learning, we believe that further improvements may lead to unsupervised features that can outperform supervised ones across many tasks of interest to the vision community.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015. 3

[2] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *arXiv preprint arXiv:1606.07419*, 2016. 3

[3] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017. 3

[4] R. Arandjelovic and A. Zisserman. Objects that sound. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451, 2018. 3

[5] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 4

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. 3

[7] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 4

[8] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7

[9] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Leveraging large-scale uncurated data for unsupervised pre-training of visual features. 2019. 7

[10] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 1

[11] S. Chopra. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Compter Vision and Pattern Recognition*, pages 539–546, 2005. 3

[12] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017. 2

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pretraining of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 2

[14] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2, 4, 5, 7

[15] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017. 3, 4, 5, 7, 9

[16] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[17] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 2, 5, 6, 7

[18] D. Elizondo. The linear separability problem: Some testing methods. *Trans. Neur. Netw.*, 17(2):330–344, Mar. 2006. 5

[19] P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991. 2

[20] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982. 2

[21] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2, 6

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[23] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 1, 3, 7

[24] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 4

[25] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[27] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158, 1995. 2

[28] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 2

[29] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio. Learning deep representations

by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 3

[30] A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016. 3

[31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[32] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015. 3

[33] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019. 2

[34] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014. 2

[35] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[36] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. *CoRR*, abs/1901.09005, 2019. 3, 5

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[38] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, pages 6874–6883, 2017. 2

[39] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 1, 3

[40] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. In *CVPR*, 2016. 3

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 3

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. 2

[43] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 3

[44] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3, 7

[45] A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273, 2013. 4

[46] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2

[47] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017. 2

[48] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996. 2

[49] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3, 4, 5

[50] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. *arXiv preprint arXiv:1612.06370*, 2016. 3, 5, 7

[51] D. Pfau, S. Petersen, A. Agarwal, D. Barrett, and K. L. Stachenfeld. Spectral inference networks: Unifying spectral methods with deep learning. *arXiv preprint arXiv:1806.02215*, 2018. 2

[52] L. Pinto, J. Davidson, and A. Gupta. Supervision via competition: Robot adversaries for learning tasks. *arXiv preprint arXiv:1610.01685*, 2016. 3

[53] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, 2016. 3

[54] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015. 2

[55] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 5

[56] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014. 2

[57] E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. *In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997. 3

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[59] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016. 3

[60] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018. 3

[61] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 3

[62] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 3

[63] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002. 2

[64] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2, 6

[65] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. Unsupervised Data Augmentation. *arXiv e-prints*, page arXiv:1904.12848, Apr 2019. 3, 6, 7

[66] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 3

[67] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. Generic 3D representation via pose estimation and matching. In *ECCV*, 2016. 3

[68] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S^4L: Self-supervised semi-supervised learning. *arXiv preprint arXiv:1905.03670*, 2019. 6, 7

[69] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2, 5, 7

[70] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009. 1

[71] X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 3