Data-Efficient Image Recognition with Contrastive Predictive Coding

Olivier J. Hénaff [1] **Aravind Srinivas** [2] **Jeffrey De Fauw** [1] **Ali Razavi** [1]
Carl Doersch [1] **S. M. Ali Eslami** [1] **Aaron van den Oord** [1]

Abstract

Human observers can learn to recognize new categories of images from a handful of examples, yet doing so with artificial ones remains an open challenge. We hypothesize that data-efficient recognition is enabled by representations which make the variability in natural signals more predictable. We therefore revisit and improve Contrastive Predictive Coding, an unsupervised objective for learning such representations. This new implementation produces features which support state-of-the-art linear classification accuracy on the ImageNet dataset. When used as input for non-linear classification with deep neural networks, this representation allows us to use 2–5× less labels than classifiers trained directly on image pixels. Finally, this unsupervised representation substantially improves transfer learning to object detection on the PASCAL VOC dataset, surpassing fully supervised pre-trained ImageNet classifiers.

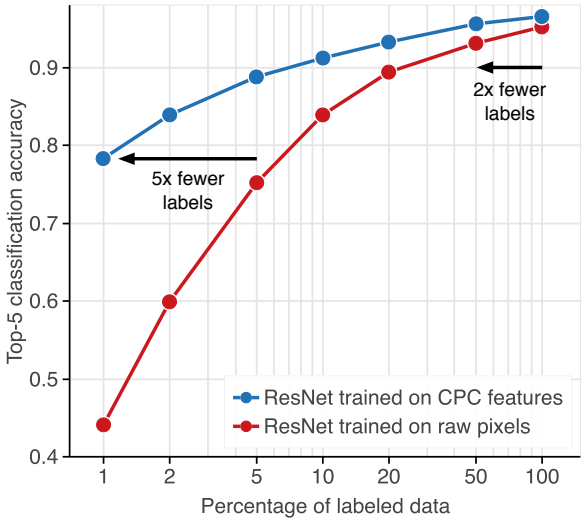

Figure 1. Data-efficient image recognition with Contrastive Predictive Coding. With decreasing amounts of labeled data, supervised networks trained on pixels fail to generalize (red). When trained on unsupervised representations learned with CPC, these networks retain a much higher accuracy in this low-data regime (blue). Equivalently, the accuracy of supervised networks can be matched with significantly fewer labels (horizontal arrows).

1. Introduction

Deep neural networks excel at perceptual tasks when labeled data are abundant, yet their performance degrades substantially when provided with limited supervision (Fig. 1, red). In contrast, humans and animals can learn about new classes of images from a small number of examples (Landau et al., 1988; Markman, 1989). What accounts for this monumental difference in data-efficiency between biological and machine vision? While highly structured representations (e.g. as proposed by Lake et al. (2015)) may improve data-efficiency, it remains unclear how to program explicit structures that capture the enormous complexity of real-world visual scenes, such as those present in the ImageNet dataset (Russakovsky et al., 2015). An alternative hypothesis has therefore proposed that intelligent systems need not be structured *a priori*, but can instead learn about the structure of the world in an unsupervised manner (Barlow, 1989; Hinton et al., 1999; LeCun et al., 2015). Choosing an appropriate training objective is an open problem, but a potential guiding principle is that useful representations should make the variability in natural signals more predictable (Tishby et al., 1999; Wiskott & Sejnowski, 2002; Richthofer & Wiskott, 2016). Indeed, human perceptual representations have been shown to linearize (or 'straighten') the temporal transformations found in natural videos, a property lacking from current supervised image recognition models (Hénaff et al., 2019), and theories of both spatial and temporal predictability have succeeded in describing properties of early visual areas (Rao & Ballard, 1999; Palmer et al., 2015). In this work, we hypothesize that spatially predictable representations may allow artificial systems to benefit from human-like data-efficiency.

Contrastive Predictive Coding (CPC, van den Oord et al. (2018)) is an unsupervised objective which learns predictable representations. CPC is a general technique that only requires in its definition that observations be ordered

[1]DeepMind, London, UK [2]University of California, Berkeley. Correspondence to: Olivier J. Hénaff <henaff@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

along e.g. temporal or spatial dimensions, and as such has been applied to a variety of different modalities including speech, natural language and images. This generality, combined with the strong performance of its representations in downstream linear classification tasks, makes CPC a promising candidate for investigating the efficacy of predictable representations for data-efficient image recognition.

Our work makes the following contributions:

- We revisit CPC in terms of its architecture and training methodology, and arrive at a new implementation with a dramatically-improved ability to linearly separate image classes (from 48.7% to 71.5% Top-1 ImageNet classification accuracy, a 23% absolute improvement), setting a new state-of-the-art.
- We then train deep neural networks on top of the resulting CPC representations using very few labeled images (e.g. 1% of the ImageNet dataset), and demonstrate test-time classification accuracy far above networks trained on raw pixels (78% Top-5 accuracy, a 34% absolute improvement), outperforming all other semi-supervised learning methods (+20% Top-5 accuracy over the previous state-of-the-art (Zhai et al., 2019)). This gain in accuracy allows our classifier to surpass supervised ones trained with 5× more labels.
- Surprisingly, this representation also surpasses supervised ResNets when given the entire ImageNet dataset (+3.2% Top-1 accuracy). Alternatively, our classifier is able to match fully-supervised ones while only using half of the labels.
- Finally, we assess the generality of CPC representations by transferring them to a new task and dataset: object detection on PASCAL VOC 2007. Consistent with the results from the previous sections, we find CPC to give state-of-the-art performance in this setting (76.6% mAP), surpassing the performance of supervised pre-training (+2% absolute improvement).

2. Experimental Setup

We first review the CPC architecture and learning objective in section 2.1, before detailing how we use its resulting representations for image recognition tasks in section 2.2.

2.1. Contrastive Predictive Coding

Contrastive Predictive Coding as formulated in (van den Oord et al., 2018) learns representations by training neural networks to predict 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 position from those above it (Fig. 2, left). These predictions are evaluated using a contrastive loss (Chopra et al., 2005; Hadsell et al., 2006), in which the network must correctly classify 'future' representations among 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 CPC architecture, each input image is first divided into a grid of overlapping patches $\boldsymbol{x}_{i,j}$, where i, j denote the location of the patch. Each patch is encoded with a neural network f_θ into a single vector $\boldsymbol{z}_{i,j} = f_\theta(\boldsymbol{x}_{i,j})$. To make predictions, a masked convolutional network g_ϕ is then applied to the grid of feature vectors. The masks are such that the receptive field of each resulting *context vector* $c_{i,j}$ only includes feature vectors that lie above it in the image (i.e. $c_{i,j} = g_\phi(\{\boldsymbol{z}_{u,v}\}_{u \leq i,v})$). The prediction task then consists of 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 $\hat{\boldsymbol{z}}_{i+k,j} = \boldsymbol{W}_k \boldsymbol{c}_{i,j}$.

The quality of this prediction is then evaluated using a contrastive loss. Specifically, the goal is to correctly recognize the target $z_{i+k,j}$ among a set of randomly sampled feature vectors $\{z_l\}$ from the dataset. We compute the probability assigned to the target using a softmax, and rate this probability using the usual cross-entropy loss. Summing this loss over locations and prediction offsets, we arrive at the CPC objective as defined in (van den Oord et al., 2018):

$$\begin{aligned}\mathcal{L}_{\text{CPC}} &= -\sum_{i,j,k} \log p(\boldsymbol{z}_{i+k,j} | \hat{\boldsymbol{z}}_{i+k,j}, \{\boldsymbol{z}_l\}) \\ &= -\sum_{i,j,k} \log \frac{\exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_{i+k,j})}{\exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_{i+k,j}) + \sum_l \exp(\hat{\boldsymbol{z}}_{i+k,j}^T \boldsymbol{z}_l)}\end{aligned}$$

The *negative samples* $\{\boldsymbol{z}_l\}$ are taken from other locations in the image and other images in the mini-batch. This loss is called InfoNCE as it is inspired by Noise-Contrastive Estimation (Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013) and has been shown to maximize the mutual information between $\boldsymbol{c}_{i,j}$ and $\boldsymbol{z}_{i+k,j}$ (van den Oord et al., 2018).

2.2. Evaluation protocol

Having trained an encoder network f_θ, a context network g_ϕ, and a set of linear predictors $\{\boldsymbol{W}_k\}$ using the CPC objective, we use the encoder to form a representation $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ of new observations $\boldsymbol{x}$, and discard the rest. Note that while pre-training required that the encoder be applied to patches, for downstream recognition tasks we can apply it directly to the entire image. We train a model h_ψ to classify these representations: given a dataset of N unlabeled images $\mathbb{D}_u = \{x_n\}$, and a (potentially much smaller) dataset of M

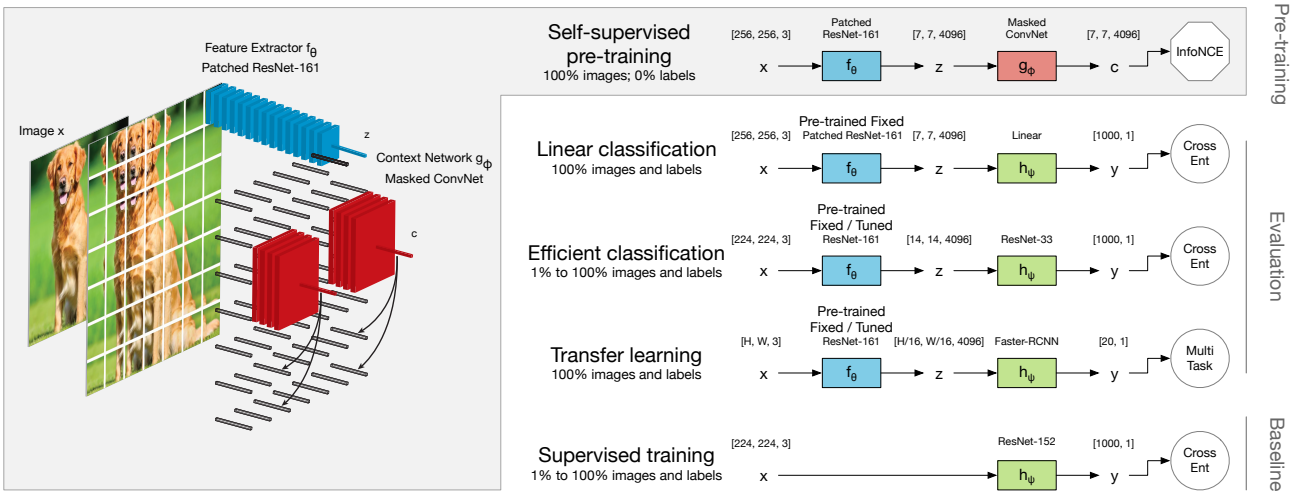

Figure 2. Overview of the framework for semi-supervised learning with Contrastive Predictive Coding. Left: unsupervised pre-training with the spatial prediction task (See Section 2.1). First, an image is divided into a grid of overlapping patches. Each patch is encoded independently from the rest with 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 (red), yielding a row of context vectors which are used to linearly predict features vectors below. Right: using the CPC representation for a classification task. Having trained the encoder network, the context network (red) is discarded and replaced by a classifier network (green) which can be trained in a supervised manner. In some experiments, we also fine-tune the encoder network (blue) for the classification task. When applying the encoder to cropped patches (as opposed to the full image) we refer to it as a *patched* ResNet in the figure.

labeled images $\mathbb{D}_l = \{x_m, y_m\}$

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

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

In all cases, the dataset of unlabeled images $\mathbb{D}_u$ we pre-train on is the full ImageNet ILSVRC 2012 training set (Russakovsky et al., 2015). We consider three labeled datasets $\mathbb{D}_l$ for evaluation, each with an associated classifier h_ψ and supervised loss $\mathcal{L}_{\text{Sup}}$ (see Fig. 2, right). This protocol is sufficiently generic to allow us to later compare the CPC representation to other methods which have their own means of learning a feature extractor f_θ.

Linear classification is a standard benchmark for evaluating the quality of unsupervised image representations. In this regime, the classification network h_ψ is restricted to mean pooling followed by a single linear layer, and the parameters of f_θ are kept fixed. The labeled dataset $\mathbb{D}_l$ is the entire ImageNet dataset, and the supervised loss $\mathcal{L}_{\text{Sup}}$ is standard cross-entropy. We use the same data-augmentation as in the unsupervised learning phase for training, and none at test time and evaluate with a single crop.

Efficient classification directly tests whether the CPC representation enables generalization from few labels. For this task, the classifier h_ψ is an arbitrary deep neural network (we use an 11-block ResNet architecture (He et al., 2016a) with 4096-dimensional feature maps and 1024-dimensional bottleneck layers). The labeled dataset $\mathbb{D}_l$ is a random subset of the ImageNet dataset: we investigated using 1%, 2%, 5%, 10%, 20%, 50% and 100% of the dataset. The supervised loss $\mathcal{L}_{\text{Sup}}$ is again cross-entropy. We use the same data-augmentation as during unsupervised pre-training, none at test-time and evaluate with a single crop.

Transfer learning tests the generality of the representation by applying it to a new task and dataset. For this we chose object detection on the PASCAL VOC 2007 dataset, a standard benchmark in computer vision (Everingham et al., 2007). As such $\mathbb{D}_l$ is the entire PASCAL VOC 2007 dataset (comprised of 5011 labeled images); h_ψ and $\mathcal{L}_{\text{Sup}}$ are the Faster-RCNN architecture and loss (Ren et al., 2015). In addition to color-dropping, we use the scale-augmentation from Doersch et al. (2015) for training.

For **linear classification**, we keep the feature extractor f_θ *fixed* to assess the representation in absolute terms. For **efficient classification** and **transfer learning**, we additionally explore *fine-tuning* the feature extractor for the supervised objective. In this regime, we initialize the feature extractor and classifier with the solutions θ^*, ψ^* found in the previous learning phase, and train them both 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.

3. Related Work

Data-efficient learning has typically been approached by two complementary methods, both of which seek to make use of more plentiful unlabeled data: representation learning and label propagation. The former formulates an objective to learn a feature extractor f_θ in an unsupervised manner, whereas the latter directly constrains the classifier h_ψ using the unlabeled data.

Representation learning saw early success using generative modeling (Kingma et al., 2014), but likelihood-based models have yet to generalize to more complex stimuli. Generative adversarial models have also been harnessed for representation learning (Donahue et al., 2016), and large-scale implementations have led to corresponding gains in linear classification accuracy (Donahue & Simonyan, 2019).

In contrast to generative models which require the reconstruction of observations, self-supervised techniques directly formulate tasks involving the learned representation. For example, simply asking a network to recognize the spatial layout of an image led to representations that transferred to popular vision tasks such as classification and detection (Doersch et al., 2015; Noroozi & Favaro, 2016). Other works showed that prediction of color (Zhang et al., 2016; Larsson et al., 2017) and image orientation (Gidaris et al., 2018), and invariance to data augmentation (Dosovitskiy et al., 2014) can provide useful self-supervised tasks. Beyond single images, works have leveraged video cues such as object tracking (Wang & Gupta, 2015), frame ordering (Misra et al., 2016), and object boundary cues (Li et al., 2016; Pathak et al., 2016). Non-visual information can be equally powerful: information about camera motion (Agrawal et al., 2015; Jayaraman & Grauman, 2015), scene geometry (Zamir et al., 2016), or sound (Arandjelovic & Zisserman, 2017; 2018) can all serve as natural sources of supervision.

While many of these tasks require predicting fixed quantities computed from the data, another class of *contrastive* methods (Chopra et al., 2005; Hadsell et al., 2006) formulate their objectives in the learned representations themselves. CPC is a contrastive representation learning method that maximizes the mutual information between spatially removed latent representations with InfoNCE (van den Oord et al., 2018), a loss function based on Noise-Contrastive Estimation (Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013). Two other methods have recently been proposed using the same loss function, but with different associated prediction tasks. Contrastive Multiview Coding (Tian et al., 2019) maximizes the mutual information between representations of different views of the same observation. Augmented Multiscale Deep InfoMax (AMDIM, Bachman et al. (2019)) is most similar to CPC in that it makes predictions across space, but differs in that it also predicts representations across layers in the model. Instance Discrimination is another contrastive objective which encourages representations that can discriminate between individual examples in the dataset (Wu et al., 2018).

A common alternative approach for improving data efficiency is **label-propagation** (Zhu & Ghahramani, 2002), where a classifier is trained on a subset of labeled data, then used to label parts of the unlabeled dataset. This label-propagation can either be discrete (as in pseudo-labeling, Lee (2013)) or continuous (as in entropy minimization, Grandvalet & Bengio (2005)). The predictions of this classifier are often constrained to be smooth with respect to certain deformations, such as data-augmentation (Xie et al., 2019) or adversarial perturbation (Miyato et al., 2018). Representation learning and label propagation have been shown to be complementary and can be combined to great effect (Zhai et al., 2019), hence we focus solely on representation learning in this work.

4. Results

When testing whether CPC enables data-efficient learning, we wish to use the best representative of this model class. Unfortunately, purely unsupervised metrics tell us little about downstream performance, and implementation details have been shown to matter enormously (Doersch & Zisserman, 2017; Kolesnikov et al., 2019). Since most representation learning methods have previously been evaluated using linear classification, we use this benchmark to guide a series of modifications to the training protocol and architecture (section 4.1) and compare to published results. In section 4.2 we turn to our central question of whether CPC enables data-efficient classification. Finally, in section 4.3 we investigate the generality of our results through transfer learning to PASCAL VOC 2007.

4.1. From CPC v1 to CPC v2

The overarching principle behind our new model design is to increase the scale and efficiency of the encoder architecture while also maximizing the supervisory signal we obtain from each image. At the same time, it is important to control the types of predictions that can be made across image patches, by removing low-level cues which might lead to degenerate solutions. To this end, we augment individual patches independently using stochastic data-processing techniques from supervised and self-supervised learning.

We identify four axes for model capacity and task setup that could impact the model's performance. The first axis increases model capacity by increasing depth and width, while the second improves training efficiency by introducing layer normalization. The third axis increases task complexity by making predictions in all four directions, and the fourth does so by performing more extensive patch-based augmentation.

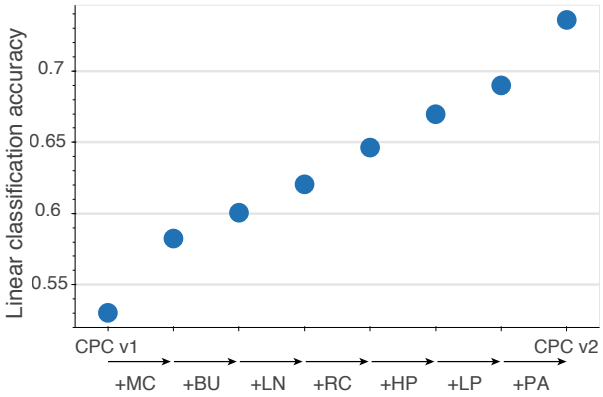

Figure 3. Linear classification performance of new variants of CPC, which incrementally add a series of modifications. MC: model capacity. BU: bottom-up spatial predictions. LN: layer normalization. RC: random color-dropping. HP: horizontal spatial predictions. LP: larger patches. PA: further patch-based augmentation. Note that these accuracies are evaluated on a custom validation set and are therefore not directly comparable to the results we report on the official validation set.

Model capacity. Recent work has shown that larger networks and more effective training improves self-supervised learning (Doersch & Zisserman, 2017; Kolesnikov et al., 2019), but the original CPC model used only the first 3 stacks of a ResNet-101 architecture. Therefore, we convert the third residual stack of the ResNet-101 (containing 23 blocks, 1024-dimensional feature maps, and 256-dimensional bottleneck layers) to use 46 blocks with 4096-dimensional feature maps and 512-dimensional bottleneck layers. We call the resulting network ResNet-161. Consistent with prior results, this new architecture delivers better performance without any further modifications (Fig. 3, **+5%** Top-1 accuracy). We also increase the model's expressivity by increasing the size of its receptive field with larger patches (from 64×64 to 80×80 pixels; **+2%** Top-1 accuracy).

Layer normalization. Large architectures are more difficult to train efficiently. Early works on context prediction with patches used batch normalization (Ioffe & Szegedy, 2015; Doersch et al., 2015) to speed up training. However, with CPC we find that batch normalization actually harms downstream performance of large models. We hypothesize that batch normalization allows these models to find a trivial solution to CPC: it introduces a dependency between patches (through the batch statistics) that can be exploited to bypass the constraints on the receptive field. Nevertheless we find that we can reclaim much of batch normalization's training efficiency by using layer normalization (**+2%** accuracy, Ba et al. (2016)).

Prediction lengths and directions. Larger architectures also run a greater risk of overfitting. We address this by asking more from the network: specifically, whereas the model in van den Oord et al. (2018) predicted each patch using only context from above, we repeatedly predict the same patch using context from below, the right and the left (using separate context networks), resulting in up to four times as many prediction tasks. Additional predictions tasks incrementally increased accuracy (adding bottom-up predictions: **+2%** accuracy; using all four spatial directions: **+2.5%** accuracy).

Table 1. Linear classification accuracy, and comparison to other self-supervised methods. In all cases the feature extractor is optimized in an unsupervised manner, using one of the methods listed below. A linear classifier is then trained on top using all labels in the ImageNet dataset, and evaluated using a single crop. Prior art reported from [1] Wu et al. (2018), [2] Zhuang et al. (2019), [3] He et al. (2019), [4] Misra & van der Maaten (2019), [5] Doersch & Zisserman (2017), [6] Kolesnikov et al. (2019), [7] van den Oord et al. (2018), [8] Donahue & Simonyan (2019), [9] Bachman et al. (2019), [10] Tian et al. (2019).

METHOD	PARAMS (M)	TOP-1	TOP-5
Methods using ResNet-50:			
INSTANCE DISCR. [1]	24	54.0	-
LOCAL AGGR. [2]	24	58.8	-
MOCO [3]	24	60.6	-
PIRL [4]	24	63.6	-
CPC V2 - RESNET-50	24	**63.8**	**85.3**
Methods using different architectures:			
MULTI-TASK [5]	28	-	69.3
ROTATION [6]	86	55.4	-
CPC V1 [7]	28	48.7	73.6
BIGBIGAN [8]	86	61.3	81.9
AMDIM [9]	626	68.1	-
CMC [10]	188	68.4	88.2
MOCO [2]	375	68.6	-
CPC V2 - RESNET-161	305	**71.5**	**90.1**

Patch-based augmentation. If the network can solve CPC using low-level patterns (e.g. straight lines continuing between patches or chromatic aberration), it need not learn semantically meaningful content. Augmenting the low-level variability across patches can remove such cues. To that effect, the original CPC model spatially jittered individual patches independently. We further this logic by adopting the 'color dropping' method of Doersch et al. (2015), which randomly drops two of the three color channels in each patch, and find it to deliver systematic gains (**+3%** accuracy). We therefore continued by adding a fixed, generic augmentation scheme using the primitives from Cubuk et al. (2018) (e.g. shearing, rotation, etc), as well as random elastic deformations and color transforms (De Fauw et al. (2018), **+4.5%**

Table 2. Data-efficient image classification. We compare the accuracy of two ResNet classifiers, one trained on the raw image pixels, the other on the proposed CPC v2 features, for varying amounts of labeled data. Note that we also fine-tune the CPC features for the supervised task, given the limited amount of labeled data. Regardless, the ResNet trained on CPC features systematically surpasses the one trained on pixels, even when given 2–5× less labels to learn from. The red (respectively, blue) boxes highlight comparisons between the two classifiers, trained with different amounts of data, which illustrate a 5× (resp. 2×) gain in data-efficiency in the low-data (resp. high-data) regime.

LABELED DATA	1%	2%	5%	10%	20%	50%	100%
				TOP-1 ACCURACY			
RESNET-200 TRAINED ON PIXELS	23.1	34.8	50.6	62.5	70.3	75.9	80.2
RESNET-33 TRAINED ON CPC FEATURES	52.7	60.4	68.1	73.1	76.7	81.2	83.4
GAIN IN DATA-EFFICIENCY	**5×**	2.5×	2×	2×	2.5×	**2×**	
				TOP-5 ACCURACY			
RESNET-200 TRAINED ON PIXELS	44.1	59.9	75.2	83.9	89.4	93.1	95.2
RESNET-33 TRAINED ON CPC FEATURES	78.3	83.9	88.8	91.2	93.3	95.6	96.5
GAIN IN DATA-EFFICIENCY	**5×**	5×	2×	2.5×	2×	**2×**	

accuracy in total). Note that these augmentations introduce some inductive bias about content-preserving transformations in images, but we do not optimize them for downstream performance (as in Cubuk et al. (2018) and Lim et al. (2019)).

Comparison to previous art. Cumulatively, these fairly straightforward implementation changes lead to a substantial improvement to the original CPC model, setting a new state-of-the-art in linear classification of 71.5% Top-1 accuracy (compared to 48.7% for the original, see table 1). Note that our architecture differs from ones used by other works in self-supervised learning, while using a number of parameters which is comparable to recently-used ones. The great diversity of network architectures (e.g. BigBiGAN employs a RevNet-50 with a $\times 4$ widening factor, AMDIM a customized ResNet architecture, CMC a ResNet-50 $\times 2$ and Momentum Contrast and ResNet-50 $\times 4$) make any apples-to-apples comparison with these works challenging. In order to compare with published results which use the same architecture, we therefore also trained a ResNet-50 architecture for the CPC v2 objective, arriving at 63.8% linear classification accuracy. This model outperforms methods which use the same architecture, as well as many recent approaches which at times use substantially larger ones (Doersch & Zisserman, 2017; van den Oord et al., 2018; Kolesnikov et al., 2019; Zhuang et al., 2019; Donahue & Simonyan, 2019).

4.2. Efficient image classification

We now turn to our original question of whether CPC can enable data-efficient image recognition.

Supervised baseline. We start by evaluating the performance of purely-supervised networks as the size of the labeled dataset $\mathbb{D}_l$ varies from 1% to 100% of ImageNet, training separate classifiers on each subset. We compared a range of different architectures (ResNet-50, -101, -152, and -200) and found a ResNet-200 to work best across all data-regimes. After tuning the supervised model for low-data classification (varying network depth, regularization, and optimization parameters) and extensive use of data-augmentation (including the transformations used for CPC pre-training), the accuracy of the best model reaches 44.1% Top-5 accuracy when trained on 1% of the dataset (compared to 95.2% when trained on the entire dataset, see Table 2 and Fig. 1, red).

Contrastive Predictive Coding. We now address our central question of whether CPC enables data-efficient learning. We follow the same paradigm as for the supervised baseline (training and evaluating a separate classifier for each labeled subset), stacking a neural network classifier on top of the CPC latents $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ rather than the raw image pixels $\boldsymbol{x}$. Specifically, we stack an 11-block ResNet classifier h_ψ on top of the 14×14 grid of CPC latents, and train it using the same protocol as the supervised baseline (see section 2.2). During an initial phase we keep the CPC feature extractor fixed and train the ResNet classifier till convergence (see Table 3 for its performance). We then fine-tune the entire stack $h_\psi \circ f_\theta$ for the supervised objective, for a small number of epochs (chosen by cross-validation). In Table 2 and Fig. 1 (blue curve) we report the results of this fine-tuned model.

This procedure leads to a substantial increase in accuracy, yielding 78.3% Top-5 accuracy with only 1% of the labels, a 34% absolute improvement (77% relative) over purely-supervised methods. Surprisingly, when given the entire dataset, this classifier reaches 83.4%/96.5% Top1/Top5 accuracy, surpassing our supervised baseline (ResNet-200:

80.2%/95.2% accuracy) and published results (original ResNet-200 v2: 79.9%/95.2%, He et al. (2016b); with AutoAugment: 80.0%/95.0%, Cubuk et al. (2018)). Using this representation also leads to gains in data-efficiency. With only 50% of the labels our classifier surpasses the supervised baseline given the entire dataset, representing a 2× gain in data-efficiency (see table 2, blue boxes). Similarly, with only 1% of the labels, our classifier surpasses the supervised baseline given 5% of the labels (i.e. a 5× gain in data-efficiency, see table 2, red boxes).

Note that we are comparing two different model *classes* as opposed to specific models or instantiations of these classes. As result we have searched for the best representative of each class, landing on the ResNet-200 for purely supervised ResNets and our wider ResNet-161 for CPC pre-training (with a ResNet-33 for downstream classification). Given the difference in capacity between these models (the ResNet-200 has approximately 60 million parameters whereas our combined model has over 500 million parameters), we verified that supervised learning would not benefit from this larger architecture. Training the ResNet-161 + ResNet-33 stack (including batch normalization throughout) in a purely supervised manner yielded results that were similar to that of the ResNet-200 (80.3%/95.2% Top-1/Top-5 accuracy). This result is to be expected: the family of ResNet-50, -101, and -200 architectures are designed for supervised learning, and their capacity is calibrated for the amount of training signal present in ImageNet labels; larger architectures only run a greater risk of overfitting. In contrast, the CPC training objective is much richer and requires larger architectures to be taken advantage of, as evidenced by the difference in linear classification accuracy between a ResNet-50 and ResNet-161 trained for CPC (table 1, 63.8% vs 71.5% Top-1 accuracy).

Other unsupervised representations. How well does the CPC representation compare to other representations that have been learned in an unsupervised manner? Table 3 compares our best model with other works on efficient recognition. We consider three objectives from different model classes: self-supervised learning with rotation prediction (Zhai et al., 2019), large-scale adversarial feature learning (BigBiGAN, Donahue & Simonyan (2019)), and another contrastive prediction objective (AMDIM, Bachman et al. (2019)). Zhai et al. (2019) evaluate the low-data classification performance of representations learned with rotation prediction using a similar paradigm and architecture (ResNet-152 with a ×2 widening factor), hence we report their results directly: given 1% of ImageNet labels, their method achieves 57.5% Top-5 accuracy. The authors of BigBiGAN and AMDIM do not report results on efficient classification, hence we evaluated these representations using the same paradigm we used for evaluating CPC. Specifically, since fine-tuned representations yield only marginal gains over fixed ones (e.g. 77.1% vs 78.3% Top-5 accuracy given 1% of the labels, see table 3), we train an identical ResNet classifier on top of these representations while keeping them fixed. Given 1% of ImageNet labels, classifiers trained on top of BigBiGAN and AMDIM achieve 55.2% and 67.4% Top-5 accuracy, respectively.

Table 3. Comparison to other methods for semi-supervised learning. *Representation learning* methods use a classifier to discriminate an unsupervised representation, and optimize it solely with respect to labeled data. *Label-propagation* methods on the other hand further constrain the classifier with smoothness and entropy criteria on unlabeled data, making the additional assumption that all training images fit into a single (unknown) testing category. When evaluating CPC v2, BigBiGAN, and AMDIM, we train a ResNet-33 on top of the representation, while keeping the representation *fixed* or allowing it to be *fine-tuned*. All other results are reported from their respective papers: [1] Zhai et al. (2019), [2] Xie et al. (2019), [3] Wu et al. (2018), [4] Misra & van der Maaten (2019).

LABELED DATA	1%	10%	100%
	TOP-5 ACCURACY		
SUPERVISED BASELINE	44.1	83.9	95.2
Methods using label-propagation:			
PSEUDOLABELING [1]	51.6	82.4	-
VAT + ENTROPY MIN. [1]	47.0	83.4	-
UNSUP. DATA AUG. [2]	-	88.5	-
ROT. + VAT + ENT. MIN. [1]	-	**91.2**	95.0
Methods using representation learning only:			
INSTANCE DISCR. [3]	39.2	77.4	-
PIRL [4]	57.2	83.8	-
ROTATION [1]	57.5	86.4	-
BIGBIGAN (FIXED)	55.2	78.8	87.0
AMDIM (FIXED)	67.4	85.8	92.2
CPC V2 (FIXED)	77.1	90.5	96.2
CPC V2 (FINE-TUNED)	**78.3**	**91.2**	**96.5**

Finally, Table 3 (top) also includes results for label-propagation algorithms. Note that the comparison is imperfect: these methods have an advantage in assuming that all unlabeled images can be assigned to a single category. At the same time, prior works (except for Zhai et al. (2019) which use a ResNet-50 ×4) report results with smaller networks, which may degrade performance relative to ours. Overall, we find that our results are on par with or surpass even the strongest such results (Zhai et al., 2019), even though this work combines a variety of techniques (entropy minimization, virtual adversarial training, self-supervised learning, and pseudo-labeling) with a large architecture whose capacity is similar to ours.

In summary, we find that CPC provides gains in data-efficiency that were previously unseen from representation learning methods, and rival the performance of the more elaborate label-propagation algorithms.

4.3. Transfer learning: image detection on PASCAL VOC 2007

We next investigate transfer learning performance on object detection on the PASCAL VOC 2007 dataset, which reflects the practical scenario where a representation must be trained on a dataset with different statistics than the dataset of interest. This dataset also tests the efficiency of the representation as it only contains 5011 labeled images to train from. The standard protocol in this setting is to train an ImageNet classifier in a supervised manner, and use it as a feature extractor for a Faster-RCNN object detection architecture (Ren et al., 2015). Following this procedure, we obtain 74.7% mAP with a ResNet-152 (Table 4). In contrast, if we use our CPC encoder as a feature extractor in the same setup, we obtain 76.6% mAP. This represents one of the first results where unsupervised pre-training surpasses supervised pre-training for transfer learning. Note that consistently with the previous section, we limit ourselves to comparing the two model *classes* (supervised vs. self-supervised), choosing the best architecture for each. Concurrently with our results, He et al. (2019) achieve 74.9% in the same setting.

Table 4. Comparison of PASCAL VOC 2007 object detection accuracy to other transfer methods. The supervised baseline learns from the entire labeled ImageNet dataset and fine-tunes for PASCAL detection. The second class of methods learns from the same *unlabeled* images before transferring. The architecture column specifies the object detector (Fast-RCNN or Faster-RCNN) and the feature extractor (ResNet-50, -101, -152, or -161). All of these methods pre-train on the ImageNet dataset, except for DeeperCluster which learns from the larger, but uncurated, YFCC100M dataset (Thomee et al., 2015). All methods fine-tune on the PASCAL 2007 training set, and are evaluted in terms of mean average precision (mAP). Prior art reported from [1] Dosovitskiy et al. (2014), [2] Doersch & Zisserman (2017), [3] Pathak et al. (2016), [4] Zhang et al. (2016), [5] Doersch et al. (2015), [6] Wu et al. (2018), [7] Caron et al. (2018), [8] Caron et al. (2019), [9] Zhuang et al. (2019), [10] Misra & van der Maaten (2019) [11] He et al. (2019).

METHOD	ARCHITECTURE	MAP
Transfer using labeled data:		
SUPERVISED BASELINE	FASTER: R152	74.7
Transfer using unlabeled data:		
EXEMPLAR [1] BY [2]	FASTER: R101	60.9
MOTION SEGM. [3] BY [2]	FASTER: R101	61.1
COLORIZATION [4] BY [2]	FASTER: R101	65.5
RELATIVE POS. [5] BY [2]	FASTER: R101	66.8
MULTI-TASK [2]	FASTER: R101	70.5
INSTANCE DISCR. [6]	FASTER: R50	65.4
DEEP CLUSTER [7]	FAST: VGG-16	65.9
DEEPER CLUSTER [8]	FAST: VGG-16	67.8
LOCAL AGGREGATION [9]	FASTER: R50	69.1
PIRL [10]	FASTER: R50	73.4
MOMENTUM CONTRAST [11]	FASTER: R50	74.9
CPC V2	FASTER: R161	**76.6**

5. Discussion

We asked whether CPC could enable data-efficient image recognition, and found that it indeed greatly improves the accuracy of classifiers and object detectors when given small amounts of labeled data. Surprisingly, CPC even improves their peformance when given ImageNet-scale labels. Our results show that there is still room for improvement using relatively straightforward changes such as augmentation, optimization, and network architecture. Overall, these results open the door toward research on problems where data is naturally limited, e.g. medical imaging or robotics.

Furthermore, images are far from the only domain where unsupervised representation learning is important: for example, unsupervised learning is already a critical step in natural language processing (Mikolov et al., 2013; Devlin et al., 2018), and shows promise in domains like audio (van den Oord et al., 2018; Arandjelovic & Zisserman, 2018; 2017), video (Jing & Tian, 2018; Misra et al., 2016), and robotic manipulation (Pinto & Gupta, 2016; Pinto et al., 2016; Sermanet et al., 2018). Currently much self-supervised work builds upon tasks tailored for a specific domain (often images), which may not be easily adapted to other domains. Contrastive prediction methods, including the techniques proposed in this paper, are task agnostic and could therefore serve as a unifying framework for integrating these tasks and modalities. This generality is particularly useful given that many real-world environments are inherently multimodal, e.g. robotic environments which can have vision, audio, touch, proprioception, action, and more over long temporal sequences. Given the importance of increasing the amounts of self-supervision (via additional prediction tasks), integrating these modalities and tasks could lead to unsupervised representations which rival the efficiency and effectiveness of human ones.

References

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

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

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

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

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

Barlow, H. Unsupervised learning. *Neural Computation*, 1 (3):295–311, 1989. doi: 10.1162/neco.1989.1.3.295.

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

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Leveraging large-scale uncurated data for unsupervised pretraining of visual features. 2019.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, pp. 539–546, 2005.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'Donoghue, B., Visentin, D., et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342, 2018.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

Doersch, C. and Zisserman, A. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2051–2060, 2017.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544*, 2019.

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

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pp. 766–774, 2014.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge 2007 (voc2007) results. 2007.

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

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.

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

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

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

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Goris, R. L., and Simoncelli, E. P. Perceptual straightening of natural videos. *Nature neuroscience*, 22 (6):984–991, 2019.

Hinton, G., Sejnowski, T., Sejnowski, H., and Poggio, T. *Unsupervised Learning: Foundations of Neural Computation*. A Bradford Book. MIT Press, 1999. ISBN 9780262581684.

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

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

Jing, L. and Tian, Y. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

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

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting self-supervised visual representation learning. *CoRR*, abs/1901.09005, 2019. URL http://arxiv.org/abs/1901.09005.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Landau, B., Smith, L. B., and Jones, S. S. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

Larsson, G., Maire, M., and Shakhnarovich, G. Colorization as a proxy task for visual understanding. In *CVPR*, pp. 6874–6883, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436, 2015.

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

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

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019.

Markman, E. M. *Categorization and naming in children: Problems of induction*. mit Press, 1989.

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

Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

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

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

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

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

Palmer, S. E., Marre, O., Berry, M. J., and Bialek, W. Predictive information in a sensory population. *Proceedings of the National Academy of Sciences*, 112(22):6908–6913, 2015.

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

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

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

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2 (1):79, 1999.

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

Richthofer, S. and Wiskott, L. Predictable feature analysis. In *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, 2016. ISBN 9781509002870. doi: 10.1109/ICMLA.2015.158.

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

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

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing (University of Illinois, Urbana, IL), Vol 37, pp 368–377.*, pp. 1–16, 1999. doi: 10.1142/S0217751X10050494.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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

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

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

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised Data Augmentation. *arXiv e-prints*, art. arXiv:1904.12848, Apr 2019.

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

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

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Zhu, X. and Ghahramani, Z. Learning from labeled and unlabeled data with label propagation. In *Technical Report CMU-CALD-02-107, Carnegie Mellon University*, 2002.

Zhuang, C., Zhai, A. L., and Yamins, D. Local aggregation for unsupervised learning of visual embeddings. *arXiv preprint arXiv:1903.12355*, 2019.