Compressive Visual Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning effective visual representations that generalize well without human supervision is a fundamental problem in order to apply Machine Learning to a wide variety of tasks. Recently, two families of self-supervised methods, contrastive learning and latent bootstrapping, exemplified by SimCLR and BYOL respectively, have made significant progress. In this work, we hypothesize that adding explicit information compression to these algorithms yields better and more robust representations. We verify this by developing SimCLR and BYOL formulations compatible with the Conditional Entropy Bottleneck (CEB) objective, allowing us to both measure and control the amount of compression in the learned representation, and observe their impact on downstream tasks. Furthermore, we explore the relationship between Lipschitz continuity and compression, showing a tractable lower bound on the Lipschitz constant of the encoders we learn. As Lipschitz continuity is closely related to robustness, this provides an alternative explanation for why compressed models are more robust. Our experiments confirm our hypotheses that adding compression to SimCLR and BYOL improves the robustness of the learned representations, and applies to a wide range of domain shifts.

1 Introduction

Individuals develop mental representations of the surrounding world that generalize over different views of a *shared context*. For instance, a shared context could be the identity of an object, as it does not change when viewed from different perspectives or lighting conditions. This ability to represent views by distilling information about the *shared context* has motivated a rich body of self-supervised learning work [38, 4, 10, 21, 23, 33]. For a concrete example, we could consider an image from the ImageNet training set [42] as a shared context, and generate different views by repeatedly applying different data augmentations. Finding stable representations of a shared context corresponds to learning a minimal high-level description since not all information is relevant or persistent. This explicit requirement of learning a concise representation leads us to prefer objectives that are *compressive* and only retain the relevant information.

Recent contrastive approaches to self-supervised visual representation learning aim to learn representations that maximally capture the mutual information between two transformed views of an image [38, 4, 10, 23, 29]. The primary idea of these approaches is that this mutual information corresponds to a general shared context that is invariant to various transformations of the input, and it is assumed that such invariant features will be effective for various downstream higher-level tasks. However, although existing contrastive approaches maximize mutual information between augmented views of the same input, they do not necessarily compress away the irrelevant information from these views, or capture relevant compression in their objectives [10, 23]. As shown in [17, 18], retaining irrelevant information often leads to less stable representations and to failures in robustness and generalization, hampering the efficacy of the learned representations. An alternative state-of-the-art self-supervised learning approach is BYOL [21], which uses a slow-moving average network to learn

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

consistent, view-invariant representations of the inputs. However, it also does not explicitly compress irrelevant information in its objective.

In this work, we modify SimCLR [10], a state-of-the-art contrastive representation learning method, by explicitly adding information compression and making it compatible with the Conditional Entropy Bottleneck (CEB) [18]. Furthermore, we show how BYOL [21] representations can also be compressed using CEB. By using CEB we are able to both measure and control the amount of information compression in the learned representation [17], and observe its impact on downstream tasks. We empirically demonstrate that our compressive variants of SimCLR and BYOL, which we name C-SimCLR and C-BYOL, consistently improve accuracy and robustness to domain shifts across a number of scenarios.

Our primary contributions are:

- Reformulations of SimCLR and BYOL such that they are compatible with information-theoretic compression using the Conditional Entropy Bottleneck [17].
- An exploration of the relationship between Lipschitz continuity, SimCLR, and CEB compression, as well as a simple, tractable lower bound on the Lipschitz constant. This provides an alternative explanation, in addition to the information-theoretic view [17, 18, 1, 2], for why CEB compression improves SimCLR model robustness.
- Extensive experiments supporting our hypothesis that adding compression to self-supervised representations like SimCLR and BYOL can consistently improve their performance and robustness to domain shifts across multiple datasets.

2 Methods

In this section, we describe the components that will allow us to make distributional, compressible versions of SimCLR and BYOL. This involves switching to the Conditional Entropy Bottleneck (CEB) objective, noting that the von Mises-Fisher (vMF) distribution is the exponential family distribution that corresponds to the cosine similarity loss function, which is used by both SimCLR and BYOL, and carefully identifying the random variables and the variational distributions needed for CEB in both SimCLR and BYOL. We also note that SimCLR and CEB together encourage learning models with a smaller Lipschitz constant, although they do not explicitly enforce that the Lipschitz constant be small.

2.1 The Conditional Entropy Bottleneck

In order to test our hypothesis that compression can improve visual representation quality, we need to be able to measure and control the amount of compression in our visual representations. To achieve this, we use the Conditional Entropy Bottleneck (CEB) [17], which has been shown to improve both test accuracy and robustness in various settings for classification models [18]. CEB is an objective function in the Information Bottleneck (IB) [46] family.

Given an observation X, a prediction target Y, and a learned representation Z of X, CEB can be written:

$$CEB \equiv \min_{Z} \beta I(X;Z|Y) - I(Y;Z) \tag{1}$$

$$= \min_{Z} \beta(H(Z) - H(Z|X) - H(Z) + H(Z|Y)) - H(Y) + H(Y|Z) \tag{2}$$

$$\equiv \min_{Z} \beta(-H(Z|X) + H(Z|Y)) + H(Y|Z) \tag{3}$$

where $H(\cdot)$ and $H(\cdot|\cdot)$ denote entropy and conditional entropy respectively. We can drop the $H(Y)$ term because it is constant with respect to Z. $I(Y;Z)$ is the useful information relevant to the task, or the prediction target Y. $I(X;Z|Y)$ is the *residual information* Z captures about X when we already know Y, which we aim to minimize. Compression strength increases as β increases.

We define $e(z|x)$, called the true encoder distribution, where z is sampled from; $b(z|y)$, a variational approximation conditioned on y; $d(y|z)$, the decoder distribution which predicts y conditioned on z. As shown in [17], CEB can be bounded variationally with these distributions:

$$vCEB \equiv \min_{e(z|x),b(z|y),d(y|z)} \mathbb{E}_{x,y\sim p(x,y),z\sim e(z|x)} \beta(\log e(z|x) - \log b(z|y)) - \log d(y|z) \tag{4}$$

There is no explicit requirement that all three distributions have learned parameters. At one limit, a model's parameters can be restricted to any one of the three distributions; at the other limit, all three distributions could have learned parameters. If $e(z|x)$ has learned parameters, its distributional form may be restricted, as we must be able to take gradients through the z samples. For example, $e(z|x)$ could not generally be a mixture distribution, as sampling the mixture distribution has a discrete component, and we cannot take gradients through discrete samples. The only requirement on the distributional form of $b(z|y)$ and $d(y|z)$ is that we be able to take gradients through their log probability functions.

InfoNCE. As shown in [17], besides parameterizing $d(y|z)$, it is possible to reuse $b(z|y)$ to make a variational bound on the $H(Y|Z)$ term. As $I(Y;Z) = H(Y) - H(Y|Z)$ and $H(Y)$ is a constant with respect to Z:

$$H(Y|Z) \geq d(y|z) \equiv \mathbb{E}_{x,y\sim p(x,y),z\sim e(z|x)} \log \frac{b(z|y)}{\sum_{k=1}^{K} b(z|y_k)} \tag{5}$$

where K is the number of examples in a minibatch. Note that Eq. 5 is the the contrastive *InfoNCE* lower bound described in [38, 39], which [17] refers to as the *CatGen* decoder because it is a categorical distribution over the minibatch that approximates the generative decoder distribution.

2.2 C-SimCLR: Compressed SimCLR

The InfoNCE bound [38] enables many contrastive visual representation methods to use it to capture shared context between different views of an image as a self-supervised objective [10, 11, 23, 13, 29]. In this work, we show how to compress the SimCLR [10] model, but the method we discuss is generally applicable to other InfoNCE-based models.

SimCLR applies randomized augmentations to an image to create two different views, x and y, and encodes both of them with a shared encoder, producing representations r_x and r_y. Both r_x and r_y are l_2-normalized. The SimCLR version of InfoNCE objective has the following form:

$$L_{NCE}(z_x, z_y) = -\log \frac{e^{\frac{1}{\tau} r_y^T r_x}}{\sum_K e^{\frac{1}{\tau} r_{y_k}^T r_x}} \tag{6}$$

where τ is a temperature term and K is the number of views in a minibatch. SimCLR further makes its InfoNCE objective *bidirectional*, such that the final objective becomes

$$L_{NCE}(r_x, r_y) + L_{NCE}(r_y, r_x) = -\log \frac{e^{\frac{1}{\tau} r_y^T r_x}}{\sum_K e^{\frac{1}{\tau} r_{y_k}^T r_x}} - \log \frac{e^{\frac{1}{\tau} r_x^T r_y}}{\sum_K e^{\frac{1}{\tau} r_{x_k}^T r_y}} \tag{7}$$

We can observe that the $\exp(\frac{1}{\tau} r_{y_k}^T r_x)$ term in Eq. (6) corresponds to $b(z|y)$ in Eq. (5), where $z = r_x$ is from $e(\cdot|x)$ and r_y is a parameter for $b(\cdot|y)$. Since SimCLR has a shared encoder, this means that $e(\cdot|x)$ and $b(\cdot|y)$ have shared parameters.

von Mises-Fisher Distributional Representations. As discussed in Sec. 2.1, CEB requires explicit distributional form of $e(\cdot|x)$ and $b(\cdot|y)$ (or at least being able to sample and take gradients through samples). To satisfy this need, we will show that SimCLR's cosine similiarity loss corresponds to choosing the von Mises-Fisher (vMF) distribution for $b(\cdot|y)$ and $e(\cdot|x)$.

vMF is a distribution on the n-dimensional hyper-sphere. The probability density function is given by $f_n(z, \mu, \kappa) = C_n(\kappa) e^{\kappa \mu^T z}$, where μ and κ are denoted as the mean direction and concentration parameter respectively. We assume κ is a constant. The normalization term $C_n(\kappa)$ is a function of κ and equal to $\frac{\kappa^{n/2-1}}{(2\pi)^{n/2} I_{n/2-1}(\kappa)}$, where I_v denotes the modified Bessel function of the first kind at order v.

We observe that by setting the mean direction μ to r_y, concentration κ_b of $b(\cdot|y)$ to $1/\tau$, and r_x to z, we recover the SimCLR objective (Equation 6) in the distributional form of InfoNCE (Equation 5)

$$\frac{e^{\frac{1}{\tau} r_y^T r_x}}{\sum_K e^{\frac{1}{\tau} r_{y_k}^T r_x}} = \frac{C_n(\kappa_b) e^{\kappa_b r_y^T r_x}}{\sum_{k=1}^{K} C_n(\kappa_b) e^{\kappa_b r_{y_k}^T r_x}} = \frac{f_n(r_x, r_y, \kappa_b)}{\sum_{k=1}^{K} f_n(r_x, r_{y_k}, \kappa_b)} = \frac{b(r_x|y)}{\sum_{k=1}^{K} b(r_x|y_k)} \tag{8}$$

Since $z = r_x$ is a deterministic unit-length vector, we can view $e(\cdot|x)$ as a spherical delta distribution, which corresponds to a vMF with r_x as the mean direction and $\kappa_e \to \infty$. In compressed SimCLR, we relax the forward encoder to have non-infinite κ_e, which results in a stochastic z.

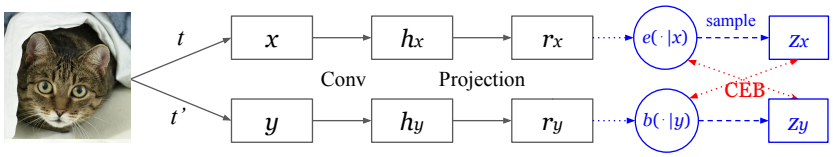

Figure 1: C-SimCLR model. t and t' denote two augmentation functions. x and y are two augmented views of the image. The upper and lower encoders' weights are shared. The blue components are our changes to SimCLR which do not add additional parameters.

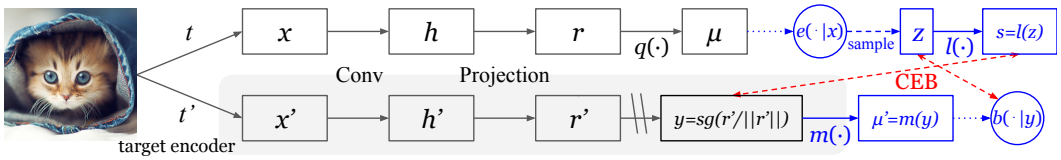

Figure 2: C-BYOL model. The upper encoder is the online encoder, and the lower is the target encoder (shaded area), which is an moving average of the online encoder (Conv + Projection). t and t' are two augmentation functions. x and x' are two augmented views of the image. sg denotes stop gradients. $q(\cdot)$ is the BYOL predictor. Blue components are our changes to BYOL. We intentionally denote the second augmented view as x' since we set the prediction target y to be $sg(r'/||r'||)$.

Compressing SimCLR with Bidirectional CEB. Figure 1 illustrates the compressed SimCLR model. As can be seen in Eq. (3), the CEB objective treats X and Y asymmetrically. However, as shown in [17], it is possible to learn a single representation Z of both X and Y by having the forward and backward encoders act as variational approximations of each other:

$$
\begin{aligned}
CEB_{\text{bidir}} &\equiv \min_{Z} \beta_X I(X;Z|Y) - I(Y;Z) + \beta_Y I(Y;Z|X) - I(X;Z) && (9)\\
&\equiv \min_{Z} \beta_X(-H(Z|X) + H(Z|Y)) + H(Y|Z) && (10)\\
&\qquad + \beta_Y(-H(Z|Y) + H(Z|X)) + H(X|Z) \\
&\leq \min_{e(z|x),b(z|y),c(y|z),d(x|z)} \mathbb{E}_{x,y\sim p(x,y)}\Big[&& (11)\\
&\qquad \mathbb{E}_{z_x\sim e(z|x)}\big[\beta_X(\log e(z_x|x) - \log b(z_x|y)) - \log c(y|z_x)\big] \\
&\qquad + \mathbb{E}_{z_y\sim b(z|y)}\big[\beta_Y(\log(b(z_y|y) - \log(e(z_y|x)) - \log d(x|z_y)\big]\Big]
\end{aligned}
$$

where $d(\cdot|\cdot)$ and $c(\cdot|\cdot)$ are the InfoNCE variational distributions of $e(\cdot|\cdot)$ and $b(\cdot|\cdot)$ respectively. Since SimCLR is trained with a bidirectional InfoNCE objective, Eq. (11) gives an easy way to compress its learned representation. As in SimCLR, the deterministic h_x (in Fig. 1) is still the representation used on downstream classification tasks.

2.3 C-BYOL: Compressed BYOL

In this section we will describe how to modify BYOL to make it compatible with CEB, as summarized in Fig. 2. BYOL [21] learns an online encoder that takes an augmented view of a given image as input and predicts outputs of a target encoder which encodes another augmented view of the image. The target encoder's parameters are updated as an exponential moving average of the online encoder's parameters. The loss function is a simple regression loss, which is equivalent to the cosine similarity between the online encoder output μ and target encoder output r' as both μ and r' are l_2-normalized:

$$
L_{byol} = ||\mu - r'||_2^2 = \mu^T\mu + r'^T r' - 2\mu^T r' = 2 - 2\mu^T r' \qquad (12)
$$

This iterative "latent bootstrapping" allows BYOL to learn a view-invariant representation that captures shared context. In contrast to SimCLR, BYOL does not rely on other samples in a batch and does not optimize the InfoNCE bound.

To make BYOL CEB-compatible, we need to identify the required random variables $x \sim X$, $y \sim Y$, $z \sim Z$, and find distributions for the encoders and decoder, $e(z|x)$, $b(z|y)$, and $d(y|z)$ as defined in Sec. 2.1. Note that BYOL is a regression task: x is the input, and the prediction target y is $r'/||r'||$.

For z and $e(z|x)$, we build a vMF distribution for $\omega = z/||z||$:

$$e_\omega(\omega|x) = f_n(\omega, \mu/||\mu||, \kappa_e) \tag{13}$$

and a Gaussian distribution for $\phi = ||z||$ independent of e_ω:

$$e_\phi(\phi|x) = N(\phi, \sigma_e^2) \tag{14}$$

Sampling ω and ϕ together gives $z = \phi\omega$, and

$$e(z|x) = e(\omega, \phi|x) = e_\omega(\omega|x)e_\phi(\phi|x) \tag{15}$$

For $d(y|z)$, we add a single linear layer l on z, which outputs $s = l(z)$, and we l_2-normalize s. We thus define

$$d(y|z) = f_n(y, \frac{s}{||s||}, \kappa_b) = C_n(\kappa_b)e^{\kappa_b y^T \frac{s}{||s||}} \tag{16}$$

Note that in the deterministic case where z is not sampled, this corresponds to adding a linear layer on μ which does not change the model capacity and empirical performance.

For $b(z|y)$, we add a 2-layer MLP on top of y and its output yields μ'. We then define a variational approximation $b(z|y) = b(\omega', \phi'|y) = b_{\omega'}(\omega'|y)b_{\phi'}(\phi'|y)$. Similar to Eq. (15), $b_{\omega'}$ is a vMF distribution of $z/||z||$ parameterized by mean direction $\mu'/||\mu'||$ and concentration κ_b, and $b_{\phi'}$ is a normal distribution of ||z|| parameterized by mean $||\mu'||$ and variance σ_b^2. These together are the components we need for making BYOL CEB-compatible and optimizing Eq. (4). As in BYOL, h is the deterministic representation we use on downstream tasks.

Finally, note that $-\log(d(y|z))$ in Eq. (4) corresponds to $-C_n(\kappa_b)e^{\kappa_b y^T \frac{s}{||s||}}$, and since $-C_n(\kappa_b)$ is a constant. When $\kappa_b = 2$, this is equivalent to the BYOL objective (Eq. (12)).

2.4 Lipschitz Continuity and Compression

Lipschitz Continuity provides a way of measuring how smooth a function is. For some function f and a distance measure $D(f(x_1), f(x_2))$, Lipschitz continuity defines an upper bound on how quickly f can change as x changes :

$$D(f(x), f(x + \Delta x)) \leq L||\Delta x||, \tag{17}$$

where L is the Lipschitz constant, Δx is the vector change in x, and $||\Delta x|| > 0$.

If we define $f(x)$ to be our encoder distribution $e(z|x)$ (which is a vMF and always positive), and the distance measure, D, to be the absolute difference of the logs of the functions, we get a Lipschitz function $L(z)$, such that:

$$\forall z \sim e(z|x), |\log e(z|x) - \log e(z|x + \Delta x)| \leq L(z)||\Delta x|| \leq L||\Delta x|| \tag{18}$$

As detailed in Sec. F, we can obtain a lower bound on the encoder *distribution*'s Lipschitz constant:

$$\frac{1}{||\Delta x||} \max\Big(\mathrm{KL}[e(z|x)||e(z|x + \Delta x)],\, \mathrm{KL}[e(z|x + \Delta x)||e(z|x)]\Big) \leq L. \tag{19}$$

To guarantee smoothness of the encoder distribution, we would like to have an upper bound on L, rather than a lower bound. Minimizing a lower bound does not directly yield any optimality guarantees relative to the bounded quantity. However, in this case, minimizing the symmetric KL below is *consistent* with learning a smoother encoder function:

$$\min_{e(z|\cdot)} \mathrm{KL}[e(z|x)||e(z|x + \Delta x)] + \mathrm{KL}[e(z|x + \Delta x)||e(z|x)] \tag{20}$$

By *consistent*, we mean that, if we could minimize this symmetric KL at every pair $(x, x + \Delta x)$ in the input domain, we would have smoothed the model. In practice, for high-dimensional input domains, that isn't possible, but minimizing Eq. (20) at a subset of the input domain still improves the model's smoothness, at least at that subset.

The minimization in Eq. (20) corresponds almost exactly to the CEB compression term in the bidirectional SimCLR models. We define $y = x + \Delta x$. At samples of the augmented observed

variables, X, Y, the C-SimCLR models minimize upper bounds on both of the following residual information terms:

$$I(X;Z|Y) + I(Y;Z|X) \leq \mathbb{E}_{x,y\sim p(x,y)} \mathrm{KL}[e(z|x)||e(z|y)] + \mathrm{KL}[e(z|y)||e(z|x)] \tag{21}$$

The only caveat to this is that we use $b(z|y)$ instead of $e(z|y)$ in C-SimCLR. b and e share weights but have different κ values in their vMF distributions. However, these are hyperparameters, so they are not part of the trained model parameters. They simply change the minimum attainable KLs in Eq. (21), thereby adjusting the minimum achievable Lipschitz constant for the models (see Sec. F).

Due to Eq. (21), we should expect that the C-SimCLR models are locally more smooth around the observed data points. We reiterate, though, that this is not a proof of increased global Lipschitz smoothness, as we are minimizing a lower bound on the Lipschitz constant, rather than minimizing an upper bound. It is still theoretically possible to learn highly non-smooth functions using CEB in this manner. It would be surprising, however, if the C-SimCLR were somehow *less* smooth than the corresponding SimCLR models.

Since various types of model robustness are related to the model's Lipschitz constant [7, 16, 48], we would expect to see that the C-SimCLR models are more robust than SimCLR models on common robustness benchmarks. It is more difficult to make the same theoretical argument for the C-BYOL models, as they do not use exactly the same encoder for both x and y. Thus, the equivalent conditional information terms from Eq. (21) are not directly minimizing a lower bound on the Lipschitz constant of the encoder. Nevertheless, we empirically explore the impact of CEB on both SimCLR and BYOL models next in Sec. 3.

3 Experimental Evaluation

We first describe our experimental set-up in Sec. 3.1, before evaluating the image representations learned by our self-supervised models in both linear evaluation settings and learning with few labels in Sec. 3.2. We then analyse the robustness and generalization of our self-supervised representations by evaluating model accuracy across a wide range of domain and distributional shifts in Sec. 3.3. Finally, we analyse the effect of compression strength in Sec. 3.4. Additional experiments and ablations can be found in the Appendix.

3.1 Experimental Set-up

Implementation details Our implementation of SimCLR, BYOL, and their compressed versions is based off of the public implementation of SimCLR [10]. The implementation stably reproduces BYOL results from [21]. We will release our code upon acceptance. We use the same set of image augmentations as in BYOL [21] for both BYOL and SimCLR, and also use BYOL's (4096, 256) two-layer projection head for both. Our SimCLR baseline reproduction results outperforms the original (details in Sec. A). We follow SimCLR and BYOL to use the LARS optimizer [49] with a cosine decay learning rate schedule [35] over 1000 epochs, with a warm-up period of 10 epochs, and scale the base learning rate 0.2 by batch size (LearningRate = 0.2 $\times$ BatchSize/256). For ablations we train for 300 epochs. As in SimCLR and BYOL, we use batch size of 4096 split over 64 Cloud TPU v3 cores. We simply follow SimCLR and BYOL in their hyperparameter choices unless otherwise stated, and provide exhaustive details in Sec. A.

Evaluation Protocol We assess the performance of representations pretrained on the ImageNet training set [42] without using any labels. Then we train a linear classifier on different labeled data on top of the frozen representation. The final performance metric is the accuracy of these classifiers. As our approach builds on SimCLR [11] and BYOL [21], we follow the same evaluation protocols. Note that we report our accuracies averaged over three pretraining trials for our main results, but omit error bars in reporting for readability as done by [21]. This is because we found the maximum absolute difference between the best and worst trials to be less than 0.15 for 1000-epoch models and 0.27 for 300 epoch models. Further details are in Sec. B.

3.2 Linear Evaluation of Self-supervised Representations

Linear evaluation on ImageNet We first evaluate the representations learned by our models by training a linear classifier on top of frozen features on the ImageNet training set, following standard

Table 1: ImageNet accuracy of linear classifier trained on representation learned with SimCLR [10] and BYOL [21], with and without CEB compression. A lower Brier score corresponds to better model calibration. We report accuracies averaged over three trials, and found the maximum absolute difference between the best and worst runs to be less than 0.27 for 300 epoch- and 0.15 for 1000 epoch-models.

	SimCLR			BYOL		
Method	Top-1	Top-5	Brier	Top-1	Top-5	Brier
ResNet-50, 300 epochs						
Uncompressed	69.0	89.1	0.420	72.9	91.2	0.371
Compressed	**70.2**	**89.8**	**0.409**	**73.6**	**91.4**	**0.364**
ResNet-50, 1000 epochs						
Uncompressed	70.6	90.1	0.400	74.2	91.7	0.356
Compressed	**71.5**	**90.6**	**0.390**	**75.0**	**92.0**	**0.347**
ResNet-50 2x, 1000 epochs						
Uncompressed	74.3	92.0	0.352	77.2	93.5	0.320
Compressed	**74.9**	**92.3**	**0.348**	**77.9**	**93.8**	**0.313**

Table 2: ImageNet accuracy when training linear classifiers with 1% and 10% of the labelled ImageNet data, on top of frozen, self-supervised representations learned on the entire ImageNet dataset without labels. For the supervised baseline, the whole ResNet-50 network is trained from random initialisation.

	Top-1		Top-5	
Method	1%	10%	1%	10%
Supervised [52]	25.4	56.4	48.4	80.4
SimCLR	49.3	63.3	75.9	85.9
C-SimCLR	**51.1**	**64.4**	**77.2**	**86.6**
BYOL	55.4	68.1	79.7	88.5
C-BYOL	**58.5**	**69.0**	**82.1**	**89.1**

Table 3: State-of-the-art comparison for linear evaluation on ImageNet. Models are trained for 1000 epochs, except for those denoted by *, which were trained for 800 epochs.

Method	Top 1	Top 5
SimCLR [10]	69.3	89.0
MoCo v2* [12]	71.1	-
SwAV* (2 crop) [9, 13]	71.8	-
InfoMin Aug* [45]	73.0	90.1
Barlow Twins [51]	73.2	91.0
BYOL [21]	74.3	91.6
C-SimCLR (ours)	71.5	90.6
C-BYOL (ours)	**75.0**	**92.0**
Supervised [10]	76.0	93.1

practice [10, 21, 31, 32]. As shown in Table 1, our compressed objectives provide consistent improvements to both SimCLR [10] and BYOL [21] across different ResNet architectures [24] of varying widths (and thus number of parameters) [50]. We note that these improvements are significant, as SimCLR and BYOL are two recent, state-of-the-art methods, and our reproduction of the SimCLR baseline outperforms that of the original paper (our uncompressed model's Top-1 accuracy is 70.6% compared to 69.3% in [10]). Similarly, our implementation of BYOL, which obtains a Top-1 of 74.2% closely matches that of [21] (74.3%).

Current self-supervised methods benefit from longer training schedules [10, 11, 21, 13, 23], and Table 1 shows that our improvements remain consistent for both 300 epoch, and the longer 1000 epoch schedule which achieves the best results. In addition to the Top-1 and Top-5 accuracies, we also compute the Brier score [6] which measures model calibration. Similar to the predictive accuracy, we observe that our compressed models obtain consistent improvements.

Learning with a few labels on ImageNet After self-supervised pretraining on ImageNet, we learn the final linear classification layer of the network on a small subset (1% and 10% respectively) of the ImageNet training set, using the class labels this time, following the standard protocol of [10, 21]. We expect that with strong feature representations, we should be able to learn an effective classifier with limited training examples.

Table 2 shows that our compressed models once again outperform their SimCLR and BYOL counterparts. The largest improvements are observed in the low-data regime, where we improve upon

Table 4: Evaluation of robustness and generalization of self-supervised models, using a ResNet-50 backbone trained on ImageNet for 1000 epochs. We report Top-1 accuracy on a range of benchmarks (detailed in Sec. 3.3 and the appendix) which measure an ImageNet-trained model's generalization to different domains and distributions.

Method	ImageNet-A	ImageNet-C	ImageNet-R	ImageNet-v2	ImageNet-Vid	YouTube-BB	ObjectNet
SimCLR	1.3	35.0	18.3	57.4	63.4	57.1	18.6
C-SimCLR	**1.5**	**36.8**	**19.6**	**59.0**	**65.3**	**58.3**	**20.8**
BYOL	1.6	42.3	24.2	62.1	67.3	61.8	23.4
C-BYOL	**1.7**	**43.8**	**24.5**	**62.9**	**68.2**	**62.1**	**23.5**

Table 5: Ablation study of CEB compression using C-SimCLR models trained for 300 epochs with a ResNet-50 backbone. β controls the level of CEB compression. We evaluate linear-evaluation on ImageNet, and model robustness on the remaining datasets as described in Sec. 3.3.

Method	ImageNet	ImageNet-A	ImageNet-C	ImageNet-R	ImageNet-v2	ImageNet-Vid	YouTube-BB	ObjectNet
SimCLR	69.0	**1.2**	32.9	17.8	56.0	61.1	58.3	17.6
$\beta = 0$	69.7	1.1	35.8	17.6	56.8	62.5	58.4	18.5
$\beta = 0.01$	69.7	**1.2**	36.2	17.5	57.2	61.2	58.5	18.7
$\beta = 0.1$	70.1	1.1	36.1	17.6	56.9	62.4	58.6	18.4
$\beta = 1$	**70.2**	1.1	**36.7**	18.2	**57.5**	**62.6**	**60.4**	**19.2**
$\beta = 1.5$	69.7	1.1	36.4	**18.3**	57.3	62.0	57.9	18.5

the state-of-the-art BYOL by 3.1% and SimCLR by 1.8%, when using only 1% of the ImageNet labels. Moreover, note that self-supervised representations significantly outperform a fully-supervised ResNet-50 baseline which overfits significantly in this low-data scenario.

State-of-the-art comparison Finally, Table 3, compares the state-of-the-art for linear evaluation accuracy on ImageNet. We present accuracies for models trained for 800 and 1000 epochs, depending on what the authors reported. In the standard setting of two augmented views, we achieve the state-of-the-art results, consistent with Table 1. We note that SwAV [9] proposed a multi-crop training strategy which improved their results. However, this technique was orthogonal to their method as it improved the results of other approaches like SimCLR [10] too. Therefore, to be consistent with all previous work, we compare results for the standard two-augmented-crop setting.

3.3 Evaluation of Model Robustness and Generalization

In this section, we analyse the robustness of our models to various domain shifts. Concretely, we use the models, with their linear classifier, from the previous experiment, and evaluate them on a suite of robustness benchmarks that have the same label set as the ImageNet dataset, as also done by [14]. As a result, we can evaluate our network and report Top-1 accuracy, as shown in Table 4, without any modifications to the network.

We consider "natural adversarial examples" with ImageNet-A [27] which consists of difficult images which a ResNet-50 classifier failed on. ImageNet-C [27] adds synthetic corruptions to the ImageNet validation set, ImageNet-R [26] considers other naturally occuring distribution changes in image style while ObjectNet [5] presents a more difficult test set for ImageNet where the authors control for different parameters such as viewpoint and background. ImageNet-Vid and YouTube-BB [43] evaluate the robustness of image classifiers to natural perturbations arising in video. Finally, ImageNet-v2 [40] is a new validation set for ImageNet where the authors attempted to replicate the original data collection process. Further details of these robustness benchmarks are in the appendix.

Table 4 shows that SimCLR and BYOL models trained with CEB compression consistently outperform their uncompressed counterparts across all seven robustness benchmarks. This is what we hypothesized in the SimCLR settings based on the Lipschitz continuity argument in Sec. 2.4 and the appendix. All models performed poorly on ImageNet-A, but this is not surprising given that ImageNet-A was collected by [27] according to images that a ResNet-50 classifier trained with full supervision on ImageNet misclassified, and we evaluate with ResNet-50 models too.

3.4 The Effect of Compression Strength

Table 5 studies the effect of the CEB compression term, β on linear evaluation accuracy on ImageNet, and also on a suite of robustness datasets. We observe that $\beta = 0$, which corresponds to no explicit

compression, but a stochastic representation, already achieves improvements across all datasets. Further improvements are observed by increasing compression (β), with $\beta = 1$ obtaining the best results. But overly strong compression can be harmful. We note that large values of β, correspond to high levels of compression, and can cause training to collapse, which we observed for $\beta = 2$.

4 Related Work

Most methods for learning visual representations without additional annotation can be roughly grouped into three families: generative, discriminative, and bootstrapping. Generative approaches build a latent embedding that models distribution over data but suffer from the expensive image generation step [47, 41, 19, 28, 30]. While early discriminative approaches used heuristic pretext tasks [15, 37], multi-view contrastive methods are among the recent state-of-the-art [10, 23, 12, 11, 38, 25, 34, 44, 9]. In this family, InfoMin Aug. [45] is closely related to ours, as it constrains the amount of mutual information between two input views by manipulating data augmentation strength. This can be viewed as a form of unconditional compression [46, 3] as it is agnostic of the prediction target, i.e. the target view in multiview contrastive learning. As discussed in [17, 18], CEB performs conditional compression that directly optimizes for the information relevant to the task, and is shown theoretically and empirically better [17, 18]. Furthermore, we showed better results than [45] in Table 3. Finally, InfoMin Aug. can also be considered complementary to our method since we compress the learning objective rather than manipulating the input data augmentation.

We also note that among the bootstrapping approaches [22, 8, 21] which BYOL [21] belongs to, BYORL [20] modified BYOL [21] to leverage Projected Gradient Descent [36] to learn a more adversarially robust encoder. The focus is, however, different from ours as we concentrate on improving generalization gap and robustness to domain shifts.

5 Conclusion

We introduced compressed versions of two state-of-the-art self-supervised algorithms, SimCLR [10] and BYOL [21], using the Conditional Entropy Bottleneck (CEB) [18]. Our extensive experiments verified our hypothesis that compressing the information content of self-supervised representations yields consistent improvements in both accuracy and robustness to domain shifts. These findings were consistent for both SimCLR and BYOL across different network backbones, datasets and training schedules. Furthermore, we presented an alternative theoretical explanation of why C-SimCLR models are more robust, in addition to the information-theoretic view [17, 18, 1, 2], by connecting Lipschitz continuity to compression.

Limitations. We note that using CEB often requires explicit and restricted distributions. This puts certain constraints on the modeling choice. It also requires additional effort to identify or create required random variables, and find appropriate distributions for them. Although we did not need additional trainable parameters for C-SimCLR, we did for C-BYOL, where we added a linear layer to the online encoder, and a 2-layer MLP to create $b(\cdot|y)$. It was, however, not difficult to observe the von Mises-Fisher distribution corresponds to loss function of BYOL and SimCLR, as well as other recent InfoNCE-based contrastive methods [9, 12, 23].

Potential Negative Societal Impact. Our work presents self-supervised methods for learning effective and robust visual representations. These representations enable learning visual classifiers with limited data (as shown by our experiments on ImageNet with 1% or 10% training data), and thus facilitates applications in many domains where annotations are expensive or difficult to collect.

Image classification systems are a generic technology with a wide range of potential applications. We are unaware of all potential applications, but are cognizant that each application has its own merits and societal impacts depending on the intentions of the individuals building and using the system. We also note that training datasets contain biases that may render models trained on them unsuitable for certain applications. It is possible that people use classification models (intentionally or not) to make decisions that impact different groups in society differently.

References

[1] Alessandro Achille and Stefano Soatto. On the emergence of invariance and disentangling in deep representations. *arXiv preprint arXiv:1706.01350*, 125:126–127, 2017.

[2] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

[3] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *NeurIPS*, 2019.

[6] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[7] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

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

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

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

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[14] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. *arXiv preprint arXiv:2007.08558*, 2020.

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[16] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George J Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *NeurIPS*, 2019.

[17] Ian Fischer. The conditional entropy bottleneck. *Entropy*, 22(9), 2020.

[18] Ian Fischer and Alexander A. Alemi. Ceb improves model robustness. *Entropy*, 22(10), 2020.

[19] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[20] Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2021.

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[22] Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, pages 3875–3886. PMLR, 2020.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

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

[25] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[27] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[28] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006.

[29] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[31] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[32] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

[33] Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *NeurIPS*, 2020.

[34] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021.

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[37] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[39] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker. On Variational Bounds of Mutual Information. *ICML*, 2019.

[40] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *ICML*, 2019.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, 2014.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[43] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? *arXiv preprint arXiv:1906.02168*, 2019.

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[45] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[46] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[47] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[48] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33, 2020.

[49] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[51] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[52] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section F
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] As mentioned in Section 3.1, we will release code and models upon acceptance. All data is already public.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.1 and Section A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 3.1. As in [21], we report our accuracies averaged over three pretraining trials for the main results but omit error bars in reporting for readability because we found the maximum absolute difference between the best and worst trials to be less than 0.15 for 1000-epoch models and 0.27 for 300 epoch models.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] As described in Section 3.1, we used Google Cloud TPU v3 accelerators, with 64 cores per experiment.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]