Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet?

Anonymous Authors[1]

Abstract

Despite recent progress made by self-supervised methods in representation learning with residual networks, they still underperform supervised learning on the ImageNet classification benchmark. To address this, we propose a novel self-supervised representation learning method Representation Learning via Invariant Causal Mechanisms v2 (ReLICv2) (based on (Mitrovic et al., 2021)) which explicitly enforces invariance over spurious features such as background and object style. We conduct an extensive experimental evaluation across a varied set of datasets, learning settings and tasks. ReLICv2 achieves 77.1% top-1 accuracy on ImageNet using linear evaluation with a ResNet50 architecture and 80.6% with larger ResNet models, outperforming previous state-of-the-art self-supervised approaches by a wide margin. Moreover, we show a relative overall improvement of exceeding +5% over the supervised baseline in the transfer setting and the ability to learn more robust representations than self-supervised and supervised models. Most notably, ReLICv2 is the first unsupervised representation learning method to consistently outperform a standard supervised baseline in a like-for-like comparison across a wide range of ResNet architectures. Finally, we show that despite using ResNet encoders, ReLICv2 is comparable to state-of-the-art self-supervised vision transformers.

1. Introduction

Learning visual representations without human supervision is an important, long-standing problem in machine learning.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022. Do not distribute.

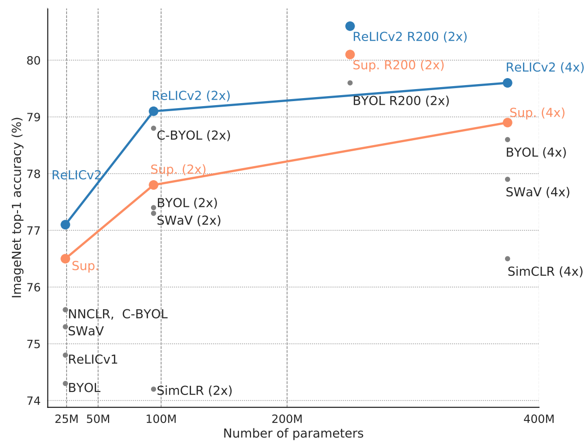

Figure 1. Top-1 linear evaluation accuracy on ImageNet using ResNet50 encoders with $1\times$, $2\times$ and $4\times$ width multipliers and a ResNet200 encoder with a $2\times$ width multiplier.

In recent years the contrastive approach to unsupervised learning has made significant strides in this direction (Chen et al., 2020a; He et al., 2019; Caron et al., 2020; Mitrovic et al., 2021). However, downstream utility[1] of these representations has until now never exceeded the performance of supervised training of the same architecture, thus limiting their usefulness.

In this work, we tackle the question "Can we outperform supervised learning without labels on ImageNet?". We hypothesize that one of the key reasons for the current subpar performance of self-supervised representations in image classification is the presence of *spurious features* such as background and object styles which have been found in the learned representations (Bordes et al., 2021). While these features are not directly informative for the task of image classification, they can be spuriously correlated with the label in the training data resulting in zero training error. Conversely, there is no guarantee that this spurious correlation will hold in the test setting; thus, encoding these spurious features in the representation can have significant negative consequences for the model's generalization performance.

To tackle this, we propose a novel self-supervised repre-

[1]This is commonly measured by how well a method performs under a standard linear evaluation protocol on ImageNet.

sentation learning method, Representation Learning via Invariant Causal Mechanisms v2 (RELICv2), which avoids encoding spurious features such as background and object style in the representation. RELICv2 achieves this by learning representations through *invariant prediction* across data which exhibits variation in object style and background. Specifically, we propose a novel fully unsupervised saliency masking method and leverage it to distinguish between semantically relevant and spurious features, i.e. foreground and background, respectively. Furthermore, we propose to use a large number of differently augmented and differently sized views of the data to learn representations that are invariant across different object styles.

We conduct an extensive experimental evaluation of our proposed method across different datasets in image classification and semantic segmentation, and across different learning settings such as transfer and out-of-distribution generalization. We also scale up RELICv2 to the Joint Foto Tree (JFT-300M) dataset (Sun et al., 2017) with 300 million images. RELICv2 achieves a new state-of-the-art performance in self-supervised learning on a wide range of ResNet architectures. On top-1 classification accuracy on ImageNet RELICv2 achieves 77.1% with a ResNet50, while with a ResNet200 $2\times$ it achieves 80.6%. Furthermore, RELICv2 is the first unsupervised representation learning method that outperforms a standard supervised baseline on linear ImageNet evaluation across ResNet50 $1\times$, $2\times$ and $4\times$ variants as well as on larger ResNet architectures such as ResNet101, ResNet152 and ResNet200; see Figure 1 and appendix for results.[2] We demonstrate the generality of RELICv2 with its competitive performance across a variety of tasks including transfer learning, semi-supervised learning, and robustness and out-of-distribution generalization. We provide further insights into how RELICv2 learns representations as well as its scaling capabilities by examining the geometry of the learned latent space in the appendix.

2. Method

RELICv2 learns representations by enforcing *invariance* over data which exhibits variability in background and object style. We obtain this data by (a) leveraging differently augmented data views of varying sizes, and (b) building a novel unsupervised saliency masking method that separates foreground from background.

Views of varying sizes. We propose to use a large number of views encoding the whole randomly augmented image as well as a small number of smaller views which contain only a portion of the randomly augmented image.[3] By explicitly enforcing invariance over this set of increasingly varied object styles, RELICv2 is able to learn representations which are increasingly invariant to the spurious features of object style. Incorporating small views which are random crops of part of the original image serves two purposes. First, as these views represent a small part of the original image, it is likely that some parts of the objects of interest might be occluded which enables us to learn representations which are more robust to object occlusions, a common issue in real-world data. Second, we hypothesize that small crops play a synergistic role to saliency masking as taking a small crop of the image is likely to remove potentially large parts of the background; see the appendix for experimental validation.

Saliency masking. To localize the semantically relevant parts of the image, we propose to use saliency masking. We develop a new fully unsupervised saliency estimation method that leverages the self-supervised refinement mechanism of DeepUSPS (Nguyen et al., 2019). In contrast to DeepUSPS, we use a ResNet50 2x network that was trained on ImageNet using a self-supervised objective as the backbone for the saliency detection networks. We also use different base handcrafted saliency methods than DeepUSPS. The pseudo-labels from each handcrafted method are refined using the self-supervision mechanism of DeepUSPS with the saliency detection network trained by fusing the refined pseudo-labels. During RELICv2 pre-training, we then randomly apply the saliency mask (computed by the saliency detection network) to the large views with a certain probability to separate the image foreground from the background. By enforcing invariance over views with the background removed, RELICv2 removes spurious background features and better captures the discriminative foreground features.

Method. Given a randomly sampled batch of datapoints $\{x_i\}_{i=1}^N$ with N the batch size, RELICv2 learns an encoder f that outputs the representation z, i.e. $z_i = f(x_i)$. Following (Chen et al., 2020a; Grill et al., 2020), we augment the input data with the data augmentation pipeline proposed in (Chen et al., 2020a) and randomly add saliency masking with a small probability; we denote the resulting augmentation pipeline with $\mathcal{T}_{sal}$. In contrast to most previous work, RELICv2 creates a large number of large views and a small number of small views[4] by randomly sampling augmentations from $\mathcal{T}_{sal}$, applying them to the input data and cropping the augmented images to the appropriate size.

RELICv2 learns representations by comparing pairs of views. Thus, let $t, t' \sim \mathcal{T}_{sal}$ yield two augmented batches $\{x_i^t\}_{i=1}^N$ and $\{x_i^{t'}\}_{i=1}^N$. Following the idea of RELIC, we

[2] Concurrent work in (Lee et al., 2021) outperforms the same standard supervised baseline only on a ResNet50 $2\times$ encoder.

[3] Most other methods use only 2 data views of the whole image.

[4] SwAV (Caron et al., 2020) propose to use small views in addition to large views, but they argue for having $3\times$ more small views than large views.

learn by maximizing the following probability

$$p(x_i^t; x_i^{t'}) = \frac{e^{\phi_\tau\left(x_i^t, x_i^{t'}\right)}}{e^{\phi_\tau\left(x_i^t, x_i^{t'}\right)} + \sum_{x_j \in \mathcal{N}(x_i)} e^{\phi_\tau\left(x_i^t, x_i^{t'}\right)}} \quad (1)$$

where $\phi_\tau(x_i, x_j) = \langle h(f(x_i)), q(g(x_j)) \rangle / \tau$ measures the similarity between embeddings with τ the temperature parameter. ReLICv2 adopts the *target* network setting of (Grill et al., 2020) such that f and g have the same architecture, but the weights of g are an exponential moving average of the weights of f; h and q are multi-layer perceptrons with h playing the role of the composition of the projector and predictor from (Grill et al., 2020) and q being the exponential moving average of the projector network. $\mathcal{N}(x_i)$ represents the set of *negatives*, i.e. datapoints with which to minimize the similarity; we construct $\mathcal{N}(x_i)$ as a small uniformly randomly sampled subset of the current batch following (Mitrovic et al., 2020).

In addition to maximizing the above probability, ReLICv2 also adopts the *invariance loss* from ReLIC defined as the Kullback-Leibler divergence between the likelihood of the two augmented views of the data as

$$D_{\mathrm{KL}}(p(x_i^t)|p(x_i^{t'})) = \mathrm{sg}\left[\mathbb{E}_{p(x_i^t; x_i^{t'})} \log p(x_i^t; x_i^{t'})\right] - \mathbb{E}_{p(x_i^t; x_i^{t'})} \log p(x_i^{t'}; x_i^t). \quad (2)$$

The invariance loss enforces that the similarity of $f(x_i^t)$ and $f(x_i^{t'})$ *relative* to the points in $\mathcal{N}(x_i)$ is the same.

Let x_i^t denote a large size view and $\tilde{x}_i^t$ be a small size view under augmentation $t \sim \mathcal{T}_{\text{sal}}$, respectively. To learn representations ReLICv2 optimizes across both large and small differently augmented views the following loss

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{1 \le l_1 \le L} \sum_{1 \le l_2 \le L} \left(-\log p(x_i^{t_{l_2}}; x_i^{t_{l_1}}) + \beta D_{\mathrm{KL}}(p(x_i^{t_{l_2}})|p(x_i^{t_{l_1}}))\right) + \sum_{1 \le s \le S} \left(-\log p(\tilde{x}_i^{t_s}; x_i^{t_{l_1}}) + \beta D_{\mathrm{KL}}(p(\tilde{x}_i^{t_s}) p(x_i^{t_{l_1}}))\right) \quad (3)$$

with $t_{l.} \sim \mathcal{T}_{\text{sal}}$ and $t_{s.} \sim \mathcal{T}$ randomly sampled data augmentations, and L and S the number of large and small views, respectively. We use the large views both for updating the encoder f as well as for computing learning targets through the target network g, i.e. $x_i^{t_{l.}}$ appears on both sides of p. On the other hand, we only use the small views for updating the encoder f and not as learning targets, i.e. $\tilde{x}_i^{s.}$ appears only on the left hand side of p, c.f. equation 1. We do not use small views as learning targets as potentially informative parts of the image might be occluded and as such the corresponding features removed from the representation. Unless otherwise noted, we use 4 large view of size 224×224 and 2 small views of size 96×96. For the precise architectural and implementation details, and related work, as well as a pseudo-code for ReLICv2 see the appendix.

Method	Top-1	Top-5
Supervised (Chen et al., 2020a)	76.5	93.7
SimCLR (Chen et al., 2020a)	69.3	89.0
MoCo v2 (Chen et al., 2020b)	71.1	-
InfoMin Aug. (Tian et al., 2020)	73.0	91.1
BYOL (Grill et al., 2020)	74.3	91.6
ReLIC (Mitrovic et al., 2021)	74.8	92.2
SwAV (Caron et al., 2020)	75.3	-
NNCLR (Dwibedi et al., 2021)	75.6	92.4
C-BYOL (Lee et al., 2021)	75.6	92.7
ReLICv2 (ours)	**77.1**	93.3

Table 1. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test for a ResNet50 encoder.

3. Experimental results

We pretrain representations without using labels on the training set of the ImageNet ILSVRC-2012 dataset (Russakovsky et al., 2015), and then extensively evaluate the learned representations in a wide variety of downstream datasets and tasks. The excellent performance of ReLICv2 across the linear evaluation, semi-supervised and transfer settings as well as the state-of-the-art scaling results on the much larger and more complex Joint Foto Tree (JFT-300M) dataset (Sun et al., 2017) showcase the generality of the approach. For a complete set of results, in particular on JFT-300M, and a detailed experimental protocol refer to the appendix. For a like-for-like comparisons with prior art (Grill et al., 2020; Caron et al., 2020; Dwibedi et al., 2021), we use as baseline the ResNet50 architecture trained with cross-entropy, a cosine learning rate schedule, full access to labels, and augmentations from (Chen et al., 2020a). More elaborate training setups have recently been proposed (Wightman et al., 2021), though they are yet to be incorporated in self-supervised models. Further analysis of the performance of ReLICv2 in terms of the class confusion, class concentration, ablation studies, importance of the invariance loss, and efficiency of representation learning is in the appendix.

Linear evaluation on ImageNet. We first evaluate ReLICv2's representations by training a linear classifier on top of the frozen encoder output according to the procedure described in (Chen et al., 2020a; Grill et al., 2020; Caron et al., 2020; Dwibedi et al., 2021) and the appendix. We report top-1 and top-5 accuracies on the ImageNet test set in Table 1. ReLICv2 outperforms all previous self-supervised approaches by a significant margin. Remarkably, ReLICv2 even outperforms a standard supervised baseline in terms of top-1 accuracy despite using no label information in pre-training. Figure 1 compares the performance of ReLICv2

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised (Zhai et al., 2019)	25.4	56.4	48.4	80.4
SimCLR (Chen et al., 2020a)	48.3	65.6	75.5	87.8
BYOL (Grill et al., 2020)	53.2	68.8	78.4	89.00
SwAV (Caron et al., 2020)	53.9	70.2	78.5	89.9
NNCLR (Dwibedi et al., 2021)	56.4	69.8	80.7	89.3
C-BYOL (Lee et al., 2021)	**60.6**	70.5	**83.4**	90.0
ReLICv2 (ours)	58.1	**72.4**	81.3	**91.2**

Table 2. Top-1 and top-5 accuracy (in %) after semi-supervised training with a fraction of ImageNet labels on a ResNet50 encoder.

against the supervised baseline and other competing methods for both the standard ResNet50 architecture as well as configurations with $2\times$ and $4\times$ wider layers and a $2\times$ wider ResNet200. ReLICv2 not only outperforms competing methods but is also the first unsupervised representation learning method which consistently outperforms the standard supervised baseline across a wide range of encoder architectures. Also, ReLICv2 outperforms the standard supervised baseline for 101, 152 and 200-layer ResNets (Grill et al., 2020) and performs competitively to the latest vision transformers (Dosovitskiy et al., 2020) at similar parameter counts. See Figure 3 and appendix for detailed results.

Semi-supervised training on ImageNet. In the semi-supervised case, representations are first pretrained, and then refined by leveraging a small subset of available labels, as per (Zhai et al., 2019; Chen et al., 2020a) among others. ReLICv2 outperforms both the standard supervised baseline and all previous self-supervised methods when using 10% of the data for fine-tuning, and performs competitively at 1% (see the appendix for detailed results).

Transfer to other tasks. We evaluate the generality of ReLICv2 representations by testing if the learned features are useful across vision tasks. For results on semantic segmentation see appendix. We perform linear evaluation and fine-tuning on the same set of classification tasks used in (Chen et al., 2020a; Grill et al., 2020; Dwibedi et al., 2021) and follow their evaluation protocol detailed in the appendix. We report standard metrics for each dataset and report performance on the held-out test set. Figure 2 compares the transfer performance of representations pre-trained using BYOL (Grill et al., 2020), NNCLR (Dwibedi et al., 2021) and ReLICv2, showing improvements over competing methods and an average relative improvement of over 5% when compared to the supervised baseline (see appendix).

Robustness and OOD generalization. To evaluate the robustness of ReLICv2 we use ImageNetV2 (Recht et al., 2019) and ImageNet-C (Hendrycks and Dietterich, 2019) datasets. For evaluating OOD generalization, we use ImageNet-R (Hendrycks et al., 2021), ImageNet-Sketch (Wang et al., 2019) and ObjectNet (Barbu et al., 2019). On

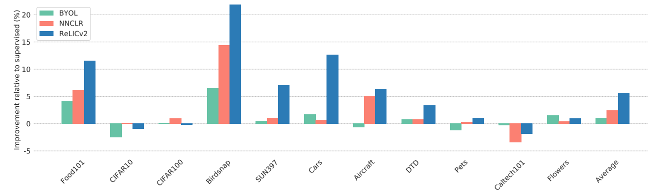

Figure 2. Transfer performance relative to the supervised baseline (a value of 0 indicates equal performance to supervised).

all datasets, we evaluate the representations from a standard ResNet50 encoder under a linear evaluation protocol, i.e. we train a linear classifier on top of the frozen representation using the labelled ImageNet training set; the test evaluation is performed zero-shot. ReLICv2 outperforms both the supervised baseline and the competing self-supervised methods on ImageNetV2 and ImageNet-C (Table 3). Also, ReLICv2 outperforms competing self-supervised methods in OOD generalization. For results and details see appendix.

Method	MF	T-0.7	Ti	IN-C
Supervised	65.1	73.9	78.4	40.9
SimCLR (Chen et al., 2020a)	53.2	61.7	68.0	31.1
BYOL (Grill et al., 2020)	62.2	71.6	77.0	42.8
ReLIC (Mitrovic et al., 2021)	63.1	72.3	77.7	44.5
ReLICv2 (ours)	**65.3**	**74.5**	**79.4**	**44.8**

Table 3. Top-1 Accuracy (in %) under linear evaluation on ImageNetV2 (matched frequency (MF), Threshold 0.7 (T-0.7) and Top Images (TI)) and ImageNet-C. ImageNet-C (IN-C) results are averaged across the 15 different corruptions.

4. Discussion

We proposed a novel self-supervised representation learning method, ReLICv2, which learns representations by enforcing invariance across background and object style. The substantial improvement over existing state-of-the-art in our extensive experimental analysis across a wide range of downstream settings, tasks and datasets highlights the usefulness of the learned representation. ReLICv2 is the first method that demonstrates that representations learned without access to labels can consistently outperform a standard supervised baseline on ImageNet which is a first step in surpassing supervised learning. Moreover, we show in the appendix that ReLICv2 outperforms recent self-supervised vision-transformer-based methods DINO (Caron et al., 2021) and MoCov3 (Chen et al., 2021) as well as exhibiting similar performance to EsViT (Li et al., 2021) for comparable parameter counts despite these methods using more powerful architectures and more involved training procedures. This suggests that combining the insights developed in ReLICv2 alongside recent architectural innovations (e.g. ViTs) could have important implications for wider adoption of self-supervised pre-training in a variety of domains as well as the design of objectives for foundational machine learning systems.

References

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *arXiv preprint arXiv:2104.13963*, 2021.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Danny Gutfreund, Joshua Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. 2019.

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2014.

Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

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.

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.

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

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

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *ArXiv*, abs/2003.04297, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *arXiv preprint arXiv:2104.14548*, 2021.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 2. Springer series in statistics New York, 2009.

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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection, 2021.

Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. Saliency detection via absorbing markov chain. In *Proceedings of the IEEE international conference on computer vision*, pages 1665–1672, 2013.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive visual representations. *arXiv preprint arXiv:2109.12909*, 2021.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *Proceedings of the IEEE international conference on computer vision*, pages 2976–2983, 2013.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019.

Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Jovana Mitrovic, Brian McWilliams, and Melanie Rey. Less can be more in contrastive learning. *PMLR*, 2020.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations (ICLR)*, 2021.

Duc Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction with self-supervision. *Advances in Neural Information Processing Systems*, 2019.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

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

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Chaitanya K Ryali, David J Schwab, and Ari S Morcos. Characterizing and improving the robustness of self-supervised learning through background augmentations. *arXiv preprint arXiv:2103.12719v2*, 2021.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pages 5628–5637, 2019.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

Yonglong Tian, C. Sun, Ben Poole, Dilip Krishnan, C. Schmid, and Phillip Isola. What makes for good views for contrastive learning. *ArXiv*, abs/2005.10243, 2020.

Yonglong Tian, Olivier J Henaff, and Aaron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. *arXiv preprint arXiv:2105.08054*, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

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

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. Distilling localization for self-supervised representation learning. *Association for the Advancement of Artificial Intelligence*, 2021.

Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821, 2014.

Wenbin Zou and Nikos Komodakis. Harf: Hierarchy-associated rich features for salient object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 406–414, 2015.

A. Comparison with vision transformers

Vision transformers (ViTs) (Dosovitskiy et al., 2020) have recently emerged as promising architectures for visual representation learning. Figure 3 compares recent ViT-based methods against RELICv2 using a variety of larger ResNet architectures. Notably, RELICv2 outperforms recent self-supervised ViT-based methods DINO (Caron et al., 2021) and MoCov3 (Chen et al., 2021) as well as exhibiting similar performance to EsViT (Li et al., 2021) for comparable parameter counts despite these methods using more powerful architectures and more involved training procedures.

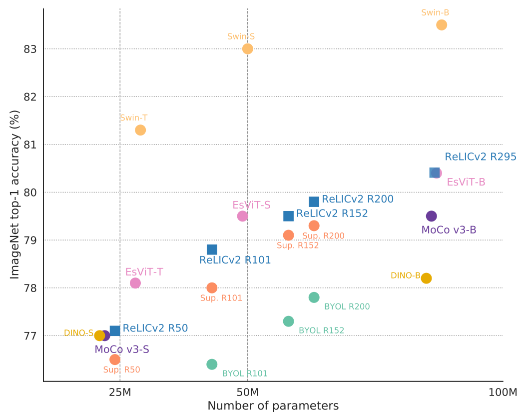

Figure 3. Comparison of ImageNet top-1 accuracy between RELICv2 and recent vision transformer-based architectures (Swin (Liu et al., 2021) is a fully supervised transformer baseline).

B. Image Preprocessing

B.1. Augmentations

Following the data augmentations protocols of (Chen et al., 2020a; Grill et al., 2020; Caron et al., 2020), RELICv2 uses a set of augmentations to generate different views of the original image which has three channels, red r, green g and blue b with $r, g, b \in [0, 1]$.

The augmentations used, in particular (corresponding to `aug` in Listing 1) are the same as in (Grill et al., 2020) and are generated as follows; for exact augmentations parameters see Table 4). The following sequence of operations is performed in the given order.

1. Crop the image: Randomly select a patch of the image, between a minimum and maximum crop area of the image, with aspect ratio sampled log-uniformly in $[3/4, 4/3]$. Upscale the patch, via bicubic interpolation, to a square image of size $s \times s$.
2. Flip the image horizontally.
3. Colour jitter: randomly adjust brightness, contrast, saturation and hue of the image, in a random order, uniformly by a value in $[-a, a]$ where a is the maximum adjustment (specified below).
4. Grayscale the image, such that the channels are combined into one channel with value $0.2989r + 0.5870g + 0.1140b$.
5. Randomly blur. Apply a 23×23 Gaussian kernel with standard deviation sampled uniformly in $[0.1, 2.0]$.
6. Randomly solarize: threshold each channel value such that all values less than 0.5 are replaced by 0 and all values above or equal to 0.5 are replaced with 1.

Apart from the initial step of image cropping, each step is executed with some probability to generate the final augmented image. These probabilities and other parameters are given in Table 4, separately for augmenting the original image x_i and the positives $\mathcal{P}(x_i)$. Note that we use 4 large views of size 224×224 pixels and 2 small views of 96×96 pixels; to get the first and third large views and the first small view we use the parameters listed below for odd views, while for the second and fourth large view and the second small view we use the parameters for even views.

Parameter	Even views	Odd views
Probability of randomly cropping	50%	50%
Probability of horizontal flip	50%	50%
Probability of colour jittering	80%	80%
Probability of grayscaling	20%	20%
Probability of blurring	100%	10%
Probability of solarization	0%	20%
Maximum adjustment a of brightness	0.4	0.4
Maximum adjustment a of contrast	0.4	0.4
Maximum adjustment a of saturation	0.2	0.2
Maximum adjustment a of hue	0.1	0.1
Crop size s	224	96 (small), 224 (large)
Crop minimum area	8%	5% (small), 14% (large)
Crop maximum area	100%	14% (small), 100% (large)

Table 4. Parameters of data augmentation scheme. Small/large indicates small or large crop.

B.2. Saliency Masking

Using unsupervised saliency masking enables us to create positives for the anchor image with the background largely removed and thus the learning process will rely less on the background to form representations. This encourages the representation to localize the objects in the image (Zhao et al., 2021).

We develop a fully unsupervised saliency estimation method that uses the self-supervised refinement mechanism from DeepUSPS (Nguyen et al., 2019) to compute saliency masks for each image in the ImageNet training set. By applying the saliency masks on top of the large views, we obtain masked images with the background removed. To further increase the background variability, instead of using a black background for the images, we apply a homogeneous grayscale to the background with the grayscale level randomly sampled for each image during training. We also use a foreground threshold such that we apply the saliency mask only if it covers at least 5% of the image. The masked images with the grayscaled background are used only during training. Specifically, with a small probability p_m we selected the masked image of the large view in place of the large view. Figure 4 shows how the saliency masks are added on top of the images to obtain the images with grayscale background.

Figure 4. Illustration of how for each image in the ImageNet training set (left) we use our unsupervised version of DeepUSPS to obtain the saliency mask (middle) which we then apply on top of the image to obtain the image with the background removed (right).

B.2.1. Training the saliency detection network to obtain saliency masks

DeepUSPS (Nguyen et al., 2019) is a saliency prediction method that uses self-supervision to refine pseudo-labels from a number of handcrafted saliency methods. To obtain saliency masks for the images in ImageNet, we build a new saliency detection method that leverages the self-supervised refinement mechanism from DeepUSPS (Nguyen et al., 2019). To this end, we firstly sample a random subset of 2500 ImageNet images; note that the original implementation of DeepUSPS uses 2500 images from the MSRA-B dataset. We instead use a randomly selected subset of the ImageNet training set of the same size to ensure a fair comparison to previous work. We compute initial saliency masks for the 2500 ImageNet images using the following handcrafted methods: Robust Background Detection (RBD) (Zhu et al., 2014), Manifold Ranking (MR) (Yang et al., 2013), Dense and Sparse Reconstruction (DSR) (Li et al., 2013) and Markov Chain (MC) (Jiang et al., 2013). Note that these methods do not make use of any supervised label information.

We then follow the two-stage mechanism proposed by DeepUSPS (Nguyen et al., 2019) to obtain a saliency prediction network. In the first stage, the noisy pseudo-labels from each handcrafted method are iteratively refined. In the second stage, these refined labels from each handcrafted saliency method are used to train the final saliency detection network. The saliency detection network is then used to compute the saliency masks for all images in the ImageNet training set. For the refinement procedure and for training the saliency detection network, we adapt the publicly available code for training DeepUSPS: https://tinyurl.com/wtlhgo3.

Note that the official implementation for DeepUSPS uses as backbone a DRN-network (Yu et al., 2017) which was pretrained on CityScapes (Cordts et al., 2016) with supervised labels. To be consistent with our fully-unsupervised setting, we replace this network with a ResNet50 2x model which was pretrained on ImageNet using the self-supervised objective from SWaV (Caron et al., 2020). We used the publicly available pretrained SWaV model from: https://github.com/facebookresearch/swav.

To account for this change in the architecture, we adjust some of the hyperparameters needed for the the two-stage mechanism of DeepUSPS. In the first stage, the pseudo-generation networks used for refining the noisy pseudo-labels from each of the handcrafted methods are trained for 25 epochs in three self-supervised iterations. We start with a learning rate of $1e-5$ which is doubled during each iteration. In the second stage, the saliency detection network is trained for 200 epochs using a learning rate of $1e-5$. We use the Adam optimizer with momentum set to 0.9 and a batch size of 10. The remaining hyperparameters are set in the same way as they are in the original DeepUSPS code.

C. RELICv2 pseudo-code in Jax

Listing 1 provides PyTorch-like pseudo-code for RELICv2 detailing how we apply the saliency masking and how the different views of data are combined in the target network setting. Note that `loss_relic` is maximizing the probability from Equation 1 and minimizing the associated KL divergence from Equation 2 between a single pair of views as proposed in (Mitrovic et al., 2021).

```
'''
f_o: online network: encoder + comparison_net
g_t: target network: encoder + comparison_net
gamma: target EMA coefficient
n_e: number of negatives
p_m: mask apply probability
'''
for x in batch: # load a batch of B samples
    # Apply saliency mask and remove background
    x_m = remove_background(x)
    for i in range(num_large_views):
        # Select either original or background-removed
        # Image with probability p_m
        x = x_m if Bernoulli(p_m) = 1 else x
        # Do large random view and augment
        xl_i = aug(crop_l(x))

        ol_i = f_o(xl_i)
        tl_i = g_t(xl_i)

    for i in range(num_small_views):
        # Do small random view and augment
        xs_i = aug(crop_s(x))
        # Small views only go through the online network
        os_i = f_o(xs_i)

    loss = 0
    # Compute loss between all pairs of large views
    for i in range(num_large_views):
        for j in range(num_large_views):
            loss += loss_relic(ol_i, tl_j, n_e)

    # Compute loss between small views and large views
    for i in range(num_small_views):
        for j in range(num_large_views):
            loss += loss_relic(os_i, tl_j, n_e)
    scale = (num_large_views + num_small_views) *
            num_large_views
    loss /= scale

    # Compute grads, update online and target networks
    loss.backward()
    update(f_o)
    g_t = gamma * g_t + (1 - gamma) * f_o
```

Listing 1. Pseudo-code for RELICv2.

D. Pretraining on ImageNet – implementation details and additional results

Similar to previous work (Chen et al., 2020a; Grill et al., 2020) we minimize our objective using the LARS optimizer (You et al., 2017) with a cosine decay learning rate schedule without restarts. Unless otherwise indicated, we train our models for 1000 epochs with a warm-up period of 10 epochs and a batch size of $|\mathcal{B}| = 4096$. In our experiments, we use 4 views of the standard size 224×224 and 2 views of the smaller size 96×96 each coming from an image augmented by a different randomly chosen data augmentation; the smaller size views are centered crops of the randomly augmented image. For a detailed ablation analysis on the number of large and small crops see Appendix G.

D.1. Linear evaluation

Following the approach of (Chen et al., 2020a; Grill et al., 2020; Caron et al., 2020; Dwibedi et al., 2021), we use the standard linear evaluation protocol on ImageNet. We train a linear classifier on top of the frozen representation which has been pretrained, i.e. the encoder parameters as well as the batch statistics are not being updated. For training the linear layer, we preprocess the data by applying standard spatial augmentations, i.e. randomly cropping the image with subsequent resizing to 224×224 and then randomly applying a horizontal flip. At test time, we resize images to 256 pixels along the shorter side with bicubic resampling and apply a 224×224 center crop to it. Both for training and testing, after performing the above processing, we normalize the color channels by substracting the average channel value and dividing by the standard deviation of the channel value (as computed on ImageNet). To train the linear classifier, we optimize the cross-entropy loss with stochastic gradient descent with Nestorov momentum for 100 epochs using a batch size of 1024 and a momentum of 0.9; we do not use any weight decay or other regularization techniques. In the following tables we report the top-1 and top-5 accuracies of different methods under a varied set of ResNet encoders of different sizes, spanning ResNet50, ResNet101, ResNet152 and ResNet200 and layer widths of $1\times$, $2\times$ and $4\times$. ResNet50 with $2\times$ and $4\times$ wider layers has 94 and 375 million parameters, respectively. ResNet101, ResNet152, ResNet200 and ResNet200 $2\times$ have 43, 58, 63 and 250 million parameters, respectively.

In the following Table 5, we present results under linear evaluation on the ImageNet test set a varied set of ResNet architectures; we compare against different unsupervised representation learning methods and use as the supervised baselines the results reported in (Chen et al., 2020a; Grill et al., 2020). Note that the supervised baselines reported in (Chen et al., 2020a) are extensively used throughout the self-supervised literature in order to compare performance against supervised learning. For architectures for which supervised baselines are not available in (Chen et al., 2020a), we use supervised baselines reported in (Grill et al., 2020) which use stronger augmentations for training supervised models than (Chen et al., 2020a) and as such do not represent a direct like-for-like comparison with self-supervised methods.

Across this varied set of ResNet architectures, ReLICv2 outperforms supervised baselines in all cases with margins up to 1.2% in absolute terms.

D.2. Semi-supervised learning

We further test ReLICv2 representations learned on bigger ResNet models in the semi-supervised setting. For this, we follow the semi-supervised protocol as in (Zhai et al., 2019; Chen et al., 2020a; Grill et al., 2020; Caron et al., 2020). First, we initialize the encoder with the parameters of the pretrained representation and we add on top of this encoder a linear classifier which is randomly initialized. Then we train both the encoder and the linear layer using either 1% or 10% of the ImageNet training data; for this we use the splits introduced in (Chen et al., 2020a) which have been used in all the methods we compare to (Grill et al., 2020; Caron et al., 2020; Dwibedi et al., 2021; Lee et al., 2021). For training, we randomly crop the image and resize it to 224×224 and then randomly apply a horizontal flip. At test time, we resize images to 256 pixels along the shorter side with bicubic resampling and apply a 224×224 center crop to it. Both for training and testing, after performing the above processing, we normalize the color channels by substracting the average channel value and dividing by the standard deviation of the channel value (as computed on ImageNet). Note that this is the same data preprocessing protocol as in the linear evaluation protocol. To train the model, we use a cross entropy loss with stochastic gradient descent with Nesterov momentum of 0.9. For both 1% and 10% settings, we train for 20 epochs and decay the initial learning rate by a factor 0.2 at 12 and 16 epochs. Following the approach of (Caron et al., 2020), we use the optimizer with different learning rates for the encoder and linear classifier parameters. For the 1% setting, we use a batch size of 2048 and base learning rates of 10 and 0.04 for the linear layer and encoder, respectively; we do not use any weight decay or other regularization technique. For the 10% setting, we use a batch size of 512 and base learning rates of 0.3 and 0.004 for the linear layer and encoder, respectively; we use a weight decay of $1e-5$, but do not use any other regularization

Method	Top-1	Top-5
Supervised (Chen et al., 2020a)	77.8	–
MoCo (He et al., 2019)	65.4	–
SimCLR (Chen et al., 2020a)	74.2	92.0
BYOL (Grill et al., 2020)	77.4	93.6
SwAV (Caron et al., 2020)	77.3	–
C-BYOL (Lee et al., 2021)	78.8	**94.5**
ReLICv2 (ours)	**79.0**	**94.5**

(a) ResNet50 2× encoder.

Method	Top-1	Top-5
Supervised (Chen et al., 2020a)	78.9	–
MoCo (He et al., 2019)	68.6	–
SimCLR (Chen et al., 2020a)	76.5	93.2
SwAV (Caron et al., 2020)	77.9	–
BYOL (Grill et al., 2020)	78.6	94.2
ReLICv2 (ours)	**79.4**	**94.3**

(b) ResNet50 4× encoder.

Method	Top-1	Top-5
Supervised (Grill et al., 2020)	78.0	94.0
BYOL (Grill et al., 2020)	76.4	9.0
ReLICv2 (ours)	**78.7**	**94.4**

(c) ResNet101 encoder.

Method	Top-1	Top-5
Supervised (Grill et al., 2020)	79.1	94.5
BYOL (Grill et al., 2020)	77.3	93.7
ReLICv2 (ours)	**79.3**	**94.6**

(d) ResNet152 encoder.

Method	Top-1	Top-5
Supervised (Grill et al., 2020)	79.3	94.6
BYOL (Grill et al., 2020)	77.8	93.9
ReLICv2 (ours)	**79.8**	**95.0**

(e) ResNet200 encoder.

Method	Top-1	Top-5
Supervised (Grill et al., 2020)	80.1	**95.2**
BYOL (Grill et al., 2020)	79.6	94.8
ReLICv2 (ours)	**80.6**	**95.2**

(f) ResNet200 2× encoder.

Table 5. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a varied set of ResNet architectures.

technique. From Table 6, we see that ReLICv2 outperforms competing self-supervised methods on ResNet50 2× in both the 1% and 10% setting. For larger ResNets, ResNet50 4× and ResNet200 2×, ReLICv2 is state-of-the-art with respect to top-1 accuracy for the low-data regime of 1%. On these networks for the higher data regime of 10% BYOL outperforms ReLICv2. Note that BYOL trains their semi-supervised models for 30 or 50 epochs whereas ReLICv2 is trained only for 20 epochs. We hypothesize that longer training (e.g. 30 or 50 epochs as BYOL) is needed for ReLICv2 representations on larger ResNets as there are more model parameters.

Method	Top-1		Top-5	
	1%	10%	1%	10%
SimCLR (Chen et al., 2020a)	58.5	71.7	83.0	91.2
BYOL (Grill et al., 2020)	62.2	73.5	84.1	91.7
ReLICv2 (ours)	**64.7**	**73.7**	**85.4**	**92.0**

(a) ResNet50 2× encoder.

Method	Top-1		Top-5	
	1%	10%	1%	10%
SimCLR (Chen et al., 2020a)	63.0	74.4	85.8	92.6
BYOL (Grill et al., 2020)	69.1	**75.7**	**87.9**	**92.5**
ReLICv2 (ours)	**69.5**	74.6	87.3	91.6

(b) ResNet50 4× encoder.

Method	Top-1		Top-5	
	1%	10%	1%	10%
BYOL (Grill et al., 2020)	71.2	77.7	**89.5**	**93.7**
ReLICv2 (ours)	**72.1**	76.4	**89.5**	93.0

(c) ResNet200 2× encoder.

Table 6. Top-1 and top-5 accuracy (in %) after semi-supervised training with a fraction of ImageNet labels for different ResNet encoders and unsupervised representation learning methods. Results are reported on the ImageNet test set.

D.3. Transfer

We follow the transfer performance evaluation protocol as outlined in (Grill et al., 2020; Chen et al., 2020a). We evaluate RELICv2 both in both transfer settings – linear evaluation and fine-tuning. For the linear evaluation protocol we freeze the encoder and train only a randomly initialized linear classifier which is put on top of the encoder. On the other hand, for fine-tuning in addition to training the randomly initialized linear classifier, we also allow for gradients to propagate to the encoder which has been initialized with the parameters of the pretrained representation. In line with prior work (Chen et al., 2020a; Grill et al., 2020; Dwibedi et al., 2021), we test RELICv2 representations on the following datasets: Food101 (Bossard et al., 2014), CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), Birdsnap (Berg et al., 2014), SUN397 (split 1) (Xiao et al., 2010), DTD (split 1) (Cimpoi et al., 2014), Cars (Krause et al., 2013) Aircraft (Maji et al., 2013), Pets (Parkhi et al., 2012), Caltech101 (Fei-Fei et al., 2004), and Flowers (Nilsback and Zisserman, 2008).

Again in line with previous methods (Chen et al., 2020a; Grill et al., 2020; Dwibedi et al., 2021), for Food101 (Bossard et al., 2014), CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), Birdsnap (Berg et al., 2014), SUN397 (split 1) (Xiao et al., 2010), DTD (split 1) (Cimpoi et al., 2014), and Cars (Krause et al., 2013) we report the Top-1 accuracy on the test set, and for Aircraft (Maji et al., 2013), Pets (Parkhi et al., 2012), Caltech101 (Fei-Fei et al., 2004), and Flowers (Nilsback and Zisserman, 2008) we report the mean per-class accuracy as the relevant metric in the comparisons. For DTD and SUN397, we only use the first split, of the 10 provided splits in the dataset as per (Chen et al., 2020a; Grill et al., 2020; Dwibedi et al., 2021).

We train on the training sets of the individual datasets and sweep over different values of the models hyperparameters. To select the best hyperparameters, we use the validation sets of the individual datasets. Using the chosen hyperparameters, we train the appropriate using the merged training and validation data and test on the held out test data in order to obtain the numbers reported in Table 7. We swept over learning rates $\{.01, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 1., 2.\}$, batch sizes $\{128, 256, 512, 1024\}$, weight decay between $\{1e{-6}, 1e{-5}, 1e{-4}, 1e{-3}, 0.01, 0.1\}$, warmup epochs $\{0, 10\}$, momentum $\{0.9, 0.99\}$, Nesterov {True, False}, and the number of training epochs. For linear transfer we considered setting epochs among $\{20, 30, 60, 80, 100\}$, and for fine-tuning, we also considered $\{150, 200, 250\}$, for datasets where lower learning rates were preferable. Models were trained with the SGD optimizer with momentum.

As can be seen from Table 7, RELICv2 representations yield better performance than both state-of-the-art self-supervised methods as well as the supervised baseline across a wide range of datasets. Specifically, RELICv2 is best on 7 out of 11 datasets and on 8 out of 11 datasets in the linear and fine-tuning settings, respectively.

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	DTD	Pets	Caltech101	Flowers
Linear evaluation:											
Supervised-IN (Chen et al., 2020a)	72.3	93.6	78.3	53.7	61.9	66.7	61.0	74.9	91.5	**94.5**	94.7
SimCLR (Chen et al., 2020a)	68.4	90.6	71.6	37.4	58.8	50.3	50.3	74.5	83.6	90.3	91.2
BYOL (Grill et al., 2020)	75.3	91.3	78.4	57.2	62.2	67.8	60.6	75.5	90.4	94.2	**96.1**
NNCLR (Dwibedi et al., 2021)	76.7	**93.7**	**79.0**	61.4	62.5	67.1	64.1	75.5	91.8	91.3	95.1
ReLICv2 (ours)	**80.6**	92.8	78.2	**65.4**	**66.2**	**75.1**	**64.8**	**77.4**	**92.4**	92.8	95.6
Fine-tuned:											
Random Init (Chen et al., 2020a)	86.9	95.9	80.2	76.1	53.6	91.4	85.9	64.8	81.5	72.6	92.0
Supervised-IN (Chen et al., 2020a)	88.3	97.5	**86.4**	75.8	64.3	92.1	86.0	74.6	92.1	93.3	97.6
SimCLR (Chen et al., 2020a)	88.2	97.7	85.9	75.9	63.5	91.3	88.1	73.2	89.2	92.1	97.0
BYOL (Grill et al., 2020)	88.5	**97.8**	86.1	76.3	63.7	91.6	88.1	76.2	91.7	**93.8**	97.0
ReLICv2 (ours)	**88.7**	97.7	85.3	**76.7**	**64.7**	**92.3**	**88.7**	**76.9**	**92.2**	93.2	**97.9**

Table 7. Accuracy (in %) of transfer performance of a ResNet50 pretrained on ImageNet.

D.4. Semantic segmentation

We evaluate the ability of RELICv2 to facilitate successful transfer of the learned representations to PASCAL (Everingham et al., 2010) and Cityscapes (Cordts et al., 2016) semantic segmentation tasks.

In accordance with (He et al., 2019), we use the RELICv2 ImageNet representation to initialise a fully convolutional backbone, which we fine-tune on the PASCAL `train_aug2012` set for 45 epochs and report the mean intersection over union (mIoU) on the `val2012` set. The fine-tuning on Cityscapes is done on the `train_fine` set for 160 epochs and evaluated on the `val_fine` set.

Method	PASCAL	Cityscapes
BYOL (Grill et al., 2020)	75.7	74.6
DetCon (Hénaff et al., 2021)	77.3	**77.0**
ReLICv2 (ours)	**77.9**	75.2

The results in the above table demonstrate that RELICv2 outperforms both BYOL and DetCon on PASCAL, reaching 77.9 IoU. RELICv2 also outperforms BYOL on Cityscapes, 75.2 vs 74.6 IoU. Note that DetCon (Hénaff et al., 2021) is a method specifically trained for detection.

D.5. Robustness and OOD Generalization

The robustness and out-of-distribution (OOD) generalization abilities of RELICv2 representations are tested on several datasets. We use ImageNetV2 (Recht et al., 2019) and ImageNet-C (Hendrycks and Dietterich, 2019) datasets to evaluate robustness. ImageNetV2 (Recht et al., 2019) has three sets of 10000 images that were collected to have a similar distribution to the original ImageNet test set, while ImageNet-C (Hendrycks and Dietterich, 2019) consists of 15 synthetically generated corruptions (e.g. blur, noise) that are added to the ImageNet test set.

For OOD generalization we examine the performance on ImageNet-R (Hendrycks et al., 2021), ImageNetSketch (Wang et al., 2019) and ObjectNet (Barbu et al., 2019). ImageNet-R (Hendrycks et al., 2021) consists of 30000 different renditions (e.g. paintings, cartoons) of 200 ImageNet classes, while ImageNet-Sketch (Wang et al., 2019) consists of 50000 images, 50 for each ImageNet class, of object sketches in the black-and-white color scheme. These datasets aim to test robustness to different textures and other naturally occurring style changes and are out-of-distribution to the ImageNet training data. ObjectNet (Barbu et al., 2019) has 18574 images from differing viewpoints and backgrounds compared to ImageNet.

On all datasets we evaluate the representations of a standard ResNet50 encoder under a linear evaluation protocol akin to Section 3, i.e. we freeze the pretrained representations and train a linear classifier using the labelled ImageNet training set; the test evaluation is performed zero-shot, i.e no training is done on the above datasets. As we've seen from Table 3, RELICv2 learns more robust representations and outperforms both the supervised baseline and the competing self-supervised methods on ImageNetV2 and ImageNet-C. We provide a detailed breakdown across the different ImageNet-C corruptions in Table 9. Furthermore, RELICv2 learns representations that outperform competing self-supervised methods while being on par with supervised performance in terms of OOD generalization; see Table 8.

Method	IN-R	IN-S	ObjectNet
Supervised	24.0	6.1	26.6
SimCLR (Chen et al., 2020a)	18.3	3.9	14.6
BYOL (Grill et al., 2020)	23.0	8.0	23.0
RELIC (Mitrovic et al., 2021)	23.8	9.1	23.8
RELICv2 (ours)	**23.9**	**9.9**	**25.9**

Table 8. Top-1 Accuracy (in %) under linear evaluation on the ImageNet-R (IN-R), ImageNet-Sketch (IN-S), ObjectNet (out-of-distribution datasets) for different unsupervised representation learning methods.

				Blur				Weather				Digital			
Method	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Supervised (Lim et al., 2019)	37.1	35.1	30.8	36.8	**25.9**	34.9	**38.1**	34.5	40.7	56.9	68.1	40.6	**45.6**	32.6	56.0
SimCLR (Chen et al., 2020a)	29.1	26.3	17.3	22.1	14.7	20.0	18.6	27.2	33.3	46.2	59.7	53.9	31.0	24.2	43.9
BYOL (Grill et al., 2020)	41.5	38.7	31.9	37.8	22.5	31.6	29.6	35.1	42.9	60.1	69.0	58.4	41.5	46.3	55.9
RELIC (Mitrovic et al., 2021)	**43.4**	**40.7**	**36.6**	**40.5**	24.5	34.3	30.5	36.6	43.8	61.4	69.5	59.5	42.8	**46.8**	57.3
RELICv2 (ours)	41.6	39.0	31.1	39.7	22.6	**35.2**	34.5	**40.1**	**46.1**	**64.5**	**71.0**	**60.0**	44.6	46.6	**58.4**

Table 9. Top-1 accuracies for for Gauss, Shot, Impulse, Blur, Weather, and Digital corruption types on ImageNet-C.

E. Pretraining on Joint Foto Tree (JFT-300M) – implementation details and additional results

E.1. Linear evaluation

We test how well ReLICv2 scales to much larger datasets by pretraining representations using the Joint Foto Tree (JFT-300M) dataset which consists of 300 million images from more than 18k classes (Hinton et al., 2015; Chollet, 2017; Sun et al., 2017). We then evaluate the learned representations on the ImageNet test set under the same linear evaluation protocol as described in section 3. We compare ReLICv2 against BYOL and Divide and Contrast (DnC) (Tian et al., 2021), a method that was specifically designed to handle large and uncurated datasets and represents the current state-of-art in self-supervised JFT-300M pretraining. Table 10 reports the top-1 accuracy when training the various methods using the standard ResNet50 architecture as the backbone for different number of ImageNet equivalent epochs on JFT-300M; implementation details can be found in the supplementary material. ReLICv2 improves over DnC by more than 2% when training on JFT for 1000 epochs and achieves better overall performance than competing methods while needing a smaller number of training epochs.

Method	Epochs	Top-1
BYOL (Grill et al., 2020)	1000	67.0
Divide and Contrast (Tian et al., 2021)	1000	67.9
ReLICv2 (ours)	1000	**70.3**
BYOL (Grill et al., 2020)	3000	67.6
Divide and Contrast (Tian et al., 2021)	3000	69.8
ReLICv2 (ours)	3000	**71.1**
BYOL (Grill et al., 2020)	5000	67.9
Divide and Contrast (Tian et al., 2021)	4500	70.7
ReLICv2 (ours)	5000	**71.4**

Table 10. Top-1 accuracy (in %) on ImageNet when learning representations using the JFT-300M dataset. Each method is pre-trained on JFT-300M for an ImageNet-equivalent number of epochs and evaluted on the ImageNet test set under a linear evaluation protocol.

For results reported in Table 10, we use the following training and evaluation protocol. To pretrain ReLICv2 on the Joint Foto Tree (JFT-300M) dataset, we used a base learning rate of 0.3 for pretraining the representations for 1000 ImageNet-equivalent epochs. For longer pretraining of 3000 and 5000 ImageNet-equivalent epochs, we use a lower base learning rate of 0.2. We set the target exponential moving average to 0.996, the contrast scale to 0.3, temperature to 0.2 and the saliency mask apply probability to 0.15 for all lenghts of pretraining. For 1000 and 5000 ImageNet-equivalent epochs we use 2.0 as the invariance scale, while for 3000 ImageNet-equivalent epochs, we use invariance scale 1.0. We then follow the linear evaluation protocol on ImageNet described in Appendix D.1. We train a linear classifier on top of the pretrained representations from JFT-300M with stochastic gradient descent with Nesterov momentum for 100 epochs using batch size of 256, learning rate of 0.5 and momentum of 0.9.

E.2. Transfer

We evaluate the transfer performance of JFT-300M pretrained representations under the linear evaluation protocol. For this, we freeze the encoder and train only linear classifier on top of the frozen encoder output, i.e. representation. As before in D.3, we follow the transfer performance evaluation protocol as outlined in (Grill et al., 2020; Chen et al., 2020a). In line with prior work, for Food101 (Bossard et al., 2014), CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), Birdsnap (Berg et al., 2014), SUN397 (split 1) (Xiao et al., 2010), DTD (split 1) (Cimpoi et al., 2014), and Cars (Krause et al., 2013) we report the top-1 accuracy on the test set, and for Aircraft (Maji et al., 2013), Pets (Parkhi et al., 2012), Caltech101 (Fei-Fei et al., 2004), and Flowers (Nilsback and Zisserman, 2008) we report the mean per-class accuracy as the relevant metric in the comparisons. For DTD and SUN397, we only use the first split, of the 10 provided splits in the dataset.

We train on the training sets of the individual datasets and sweep over different values of the models hyperparameters. To select the best hyperparameters, we use the validation sets of the individual datasets. Using the chosen hyperparameters, we train the linear layer from scratch using the merged training and validation data and test on the held out test data in order to obtain the numbers reported in Table 11. We swept over learning rates $\{.01, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 1., 2.\}$, batch sizes $\{128, 256, 512, 1024\}$, weight decay between $\{1e{-}6, 1e{-}5, 1e{-}4, 1e{-}3, 0.01, 0.1\}$, warmup epochs $\{0, 10\}$, momentum $\{0.9, 0.99\}$, Nesterov {True, False}, and the number of training epochs $\{60, 80, 100\}$. Models were trained with the SGD optimizer with momentum.

As can be seen from Table 11, longer pretraining benefits transfer performance of RELICv2. Although DnC (Tian et al., 2021) was specifically developed to handle uncurated datasets such as JFT-300M, we see that RELICv2 has comparable performance to DnC in terms of the number of datasets with state-of-the-art performance among self-supervised representation learning methods; this showcases the generality of RELICv2.

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	DTD	Pets	Caltech101	Flowers
BYOL-5k (Grill et al., 2020)	73.3	89.8	72.4	38.2	61.8	64.4	54.4	75.5	77	90.1	94.3
DnC-4.5k (Tian et al., 2021)	**78.7**	**91.7**	**74.9**	42.1	65.0	75.3	54.1	76.6	**86.1**	90.2	**98.2**
ReLICv2-1k (ours)	77.5	90.2	72.6	47.4	64.5	74.4	62.9	**77.0**	84.9	**92.2**	94.5
ReLICv2-5k (ours)	78.3	89.9	73.0	**49.4**	**65.6**	**76.9**	**65.5**	76.8	85.1	91.4	95.7

Table 11. Accuracy (in %) of transfer performance of a ResNet50 pretrained on JFT under the linear transfer evaluation protocol. xk refers to the length of pretraining in ImageNet-equivalent epochs, e.g. 1k corresponds to 1000 ImageNet-equivalent epochs of pretraining.

E.3. Robustness and OOD Generalization

We also tested the robustness and out-of-distribution (OOD) generalization of RELICv2 representations pretrained on JFT. We use the same set-up described in D.5 where we freeze the pretrained representations on JFT-300M, train a linear classifier using the labelled ImageNet training set and perform zeroshot test evaluation on datasets testing robustness and OOD generalization. As in D.5, we evaluated robustness using the ImageNetV2 (Recht et al., 2019) and ImageNet-C (Hendrycks and Dietterich, 2019) datasets and OOD generalization using ImageNet-R (Hendrycks et al., 2021), ImageNetSketch (Wang et al., 2019) and ObjectNet (Barbu et al., 2019) datasets. We report the robustness results in Table 12a and the OOD generalization results in Table 12b. We notice that RELICv2 representations pretrained on JFT-300M for different number of ImageNet-equivalent epochs have worse robustness and OOD generalization performance compared to RELICv2 representations pretrained directly on ImageNet (see Table 3 and Table 8 for reference). Given that the above datasets have been specifically constructed to measure the robustness and OOD generalization abilities of models pretrained on ImageNet (as they have been constructed in relation to ImageNet), this result is not entirely surprising. We hypothesize that this is due to there being a larger discrepancy between datasets and JFT-300M than these datasets and ImageNet and as such JFT-300M-pretrained representations perform worse than ImageNet-pretrained representations. Additionally, note that pretraining on JFT-300M for longer does not necessarily result in better downstream performance on the robustness and out-of-distribution datasets.

Epochs	MF	T-0.7	Ti	IN-C
1000	57.6	66.7	73.0	32.9
3000	58.6	67.5	73.4	32.8
5000	59.1	67.3	73.3	33.5

(a) ImageNetv2 dataset.

Epochs	IN-R	IN-Sketch	ObjectNet
1000	20.4	6.7	20.3
3000	20.3	8.7	21.3
5000	20.3	5.4	20.9

(b) ImageNet-R, ImageNet-Sketch and ObjectNet datasets.

Table 12. Top-1 Accuracy (in %) under linear evaluation on the the ImageNet-R (IN-R), ImageNet-Sketch (IN-S) and ObjectNet out-of-distribution datasets and on ImageNetV2 dataset for RELICv2 pre-trained on JFT-300M for different numbers of ImageNet-equivalent epochs. We evaluate on all three variants on ImageNetV2 – matched frequency (MF), Threshold 0.7 (T-0.7) and Top Images (TI). The results for ImageNet-C (IN-C) are averaged across the 15 different corruptions.

F. Comparison between self-supervised methods

Contrastive multi-view approaches for unsupervised representation learning have recently shown excellent performance in visual recognition tasks (Oord et al., 2018; Bachman et al., 2019; Chen et al., 2020a; He et al., 2019; Dwibedi et al., 2021; Grill et al., 2020), as did boostrapping-based multi-view learning (Grill et al., 2020). Explicitly enforcing invariance via clustering (Caron et al., 2020) or based on causal perspectives (Mitrovic et al., 2021) has been promising, the latter leading to more compact representations. The use of background augmentations has recently been gaining attention (Zhao et al., 2021; Ryali et al., 2021), though ReLICv2 utilises these across multiple views of varying sizes, and includes an invariance loss. Competitive results with the supervised baseline have recently been reported in (Lee et al., 2021), based on a conditional entropy bottleneck approach.

In this review we focus on how important algorithmic choices: namely explicitly enforcing invariance and more considered treatment of positive and negative examples are key factors in improving downstream classification performance of unsupervised representations.

Negatives. A key observation of (Chen et al., 2020a) was that large batches (up to 4096) improve results. This was partly attributed to the effect of more negatives. This motivated the incorporation of queues that function as large reservoirs of negative examples into contrastive learning (He et al., 2019). However subsequent work has shown that naively using a large number of negatives can have a detrimental effect on learning (Mitrovic et al., 2020; Saunshi et al., 2019; Chuang et al., 2020; Robinson et al., 2020). One reason for this is due to *false negatives*, that is points in the set of negatives which actually belong to the same latent class as the anchor point. These points are likely to have a high relative similarity to the anchor under ϕ and therefore contribute disproportionately to the loss. This will have the effect of pushing apart points belonging to the same class in representation space. The selection of true negatives is a difficult problem as in the absence of labels it necessitates having access to reasonably good representations to begin with. As we do not have access to these representations, but are instead trying to learn them, there has been limited success in avoiding false and selecting informative negatives. This phenomenon explains the limited success of attempts to perform hard negative sampling.

Subsampling-based approaches have been proposed to avoid false negatives via importance sampling to attempt to find *true* negatives which are close to the latent class boundary of the anchor point (Robinson et al., 2020), or uniformly-at-random sampling a small number of points to avoid false negatives (Mitrovic et al., 2020).

Positives and invariance. Learning representations which are invariant to data augmentation is known to be important for self-supervised learning. Invariance is achieved heuristically through comparing two different augmentations of the same anchor point. Incorporating an explicit clustering step is another way of enforcing some notion of invariance (Caron et al., 2020). However, neither of these strategies can be directly linked theoretically to learning more compact representations. More rigorously (Mitrovic et al., 2021) approach invariance from a causal perspective. They show that invariance must be explicitly enforced—via an invariance loss in addition to the contrastive loss—in order to obtain guaranteed generalization performance. Most recently (Dwibedi et al., 2021) and (Assran et al., 2021) use nearest neighbours to identify other elements from the batch which potentially belong to the same class as the anchor point.

Table 13 provides a detailed comparison in terms of how prominent representation learning methods utilize positive and negative examples and how they incorporate both explicit contrastive and invariance losses. Here $\text{aug}(x_i)$ refers to the standard set of SimCLR augmentations (Chen et al., 2020a), $\text{nn}(x_i)$ refers to a scheme which selects nearest neighbours of x_i, $\text{mc}(x_i)$ are multicrop augmentations (c.f. (Caron et al., 2020)). $\text{proto}^+(x_i)$ and $\text{proto}^-(x_i)$ refer to using prototypes computed via an explicit clustering step c.f. (Caron et al., 2020). Finally, $\text{sal}(x_i)$ refers to a scheme which computes saliency masks of x_i and removes backgrounds as described in section 2. Note that SwAV first computes a clustering of the batch then contrasts the embedding of the point and its nearest cluster centroid (proto^+) against the remaining $K - 1$ cluster centroids (proto^-); invariance is implicitly enforced in the clustering step.

Method	Contrastive	Invariance	Positives	Negatives
SimCLR (Chen et al., 2020a)	✓	✗	aug(x_i)	full batch
BYOL (Grill et al., 2020)	✗	ℓ_2	aug(x_i)	n/a
NNCLR (Dwibedi et al., 2021)	✓	✗	aug(x_i), nn(x_i)	full batch
MoCo (He et al., 2019)	✓	✗	aug(x_i)	queue
SwAV (Caron et al., 2020)	✓	✗	aug(x_i), mc(x_i), proto$^+(x_i)$	proto$^-(x_i)$
Debiased (Chuang et al., 2020)	✓	✗	aug(x_i)	importance sample
Hard Negatives (Robinson et al., 2020)	✓	✗	aug(x_i)	importance sample
ReLICv1 (Mitrovic et al., 2021)	✓	D_{KL}	aug(x_i)	subsample
ReLICv2 (ours)	✓	D_{KL}	aug(x_i), mc(x_i), sal(x_i)	subsample

Table 13. The role of positives and negatives in recent unsupervised representation learning algorithms.

G. Analysis

G.1. Scaling analysis

Figure 5 shows the ImageNet linear evaluation accuracy obtained by representations learned using ReLICv2 as a function of the number of images seen during pre-training using the ImageNet training set. It can be seen that in order to reach 70% accuracy the ResNet50 model requires approximately twice the number of iterations as the ResNet295 model. The ResNet295 has approximately $3.6\times$ the number of parameters as the ResNet50 (87M vs 24M, respectively). This finding is in accordance with other works which show that larger models are more sample efficient (i.e. they require fewer samples to reach a given accuracy) (Zhai et al., 2021).

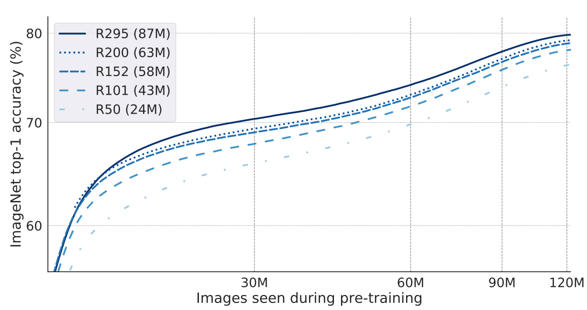

Figure 5. ImageNet accuracy obtained by ReLICv2 as a function of number of images seen during pre-training for a variety of ResNet architectures. The number of parameters of each model is in parenthesis.

G.2. Class confusion analysis

To understand the effect of the invariance term in ReLICv2, we look at the distances between learned representations of closely related classes. Figure 6 illustrates the Euclidean distances between nearest-neighbour (NN) representations learned by ReLICv2 and BYOL on ImageNet using the protocol described in section 3. Here we pick two breeds of dog and two breeds of cat. Each of these four classes has 50 points associated with it from the ImageNet test set, ordered contiguously. Each row represents an image and each coloured point in a row represents one of the five nearest neighbours of the representation of that image where the colour indicates the distance between the image and the NN. Representations which align perfectly with the underlying class structure would exhibit a perfect block-diagonal structure, i.e. their NNs all belong to the same underlying class. We see that ReLICv2 learns representations whose NNs are closer and exhibit less confusion between classes and super-classes than BYOL.

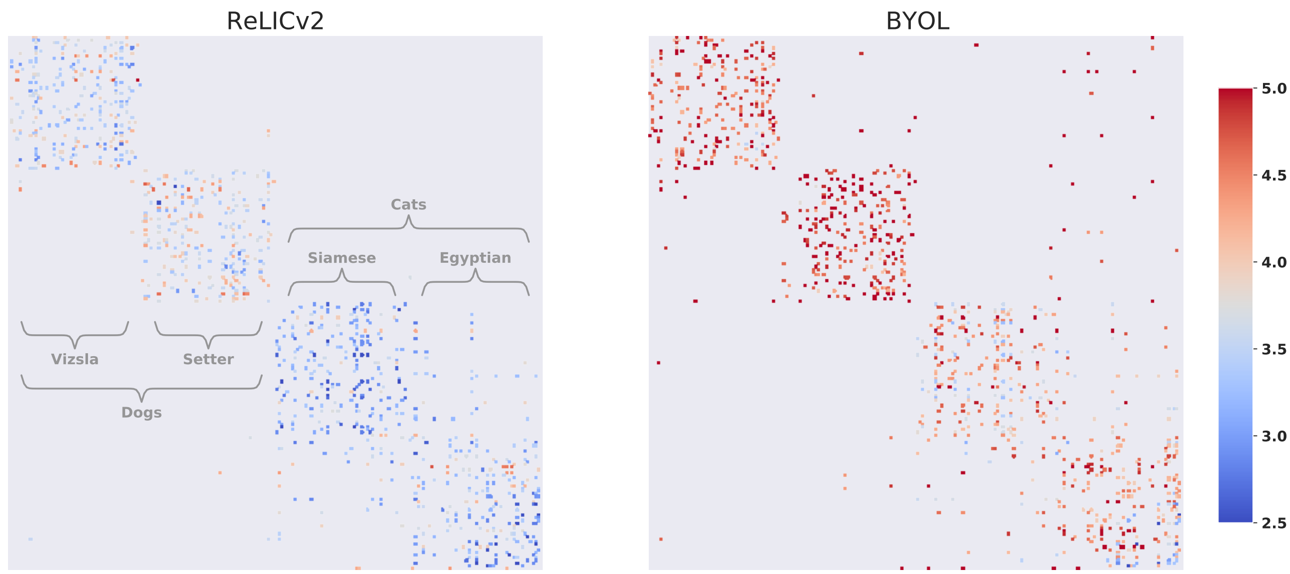

Figure 6. Distances between nearest-neighbour representations. Each coloured point in a row represents one of the five nearest neighbours of the representation of that image where the colour indicates the distance between the points.

G.3. Class concentration

To quantify the overall structure of the learned latent space, we examine the within- and between-class distances of all classes. Figure 7 compares the distribution of ratios of between-class and within-class ℓ_2-distances of the representations of points in the ImageNet test set learned by ReLICv2 against those learned by a standard supervised baseline.[5] A larger ratio implies that the representation is better concentrated within the corresponding classes and better separated between classes and therefore more easily linearly separated (c.f. Fisher's linear discriminants (Friedman et al., 2009)). We see that ReLICv2's distribution is shifted to the right (i.e. having a higher ratio) compared to the standard supervised baseline suggesting that the representations can be better separated using a linear classifier. The empirical results in this section further confirm the theoretical insights of (Mitrovic et al., 2021) and explain the superior performance of ReLICv2 reported in section 3.

G.4. Views of varying sizes

Most prior work uses 2 views of size 224×224 to learn representations, while ReLICv2 proposes the use of a larger number of views of that size combined with a few smaller views. We ablate the use of different numbers of large and small views in ReLICv2 using only standard SimCLR augmentations (i.e. without saliency masking). Below is the top-1 ImageNet test set performance under the linear evaluation protocol on a ResNet50 pretrained for 1000 epochs for different numbers of large and small views; $[L, S]$ denotes using L large views and S small views.

Views	[2, 0]	[2, 2]	[2, 6]	[4, 0]	[4, 2]	[6, 2]	[8, 2]
Top-1	74.8	76.2	76.0	75.5	**76.8**	76.5	76.5

[5] Both ReLICv2 and the standard supervised baseline were trained on the ImageNet training set.

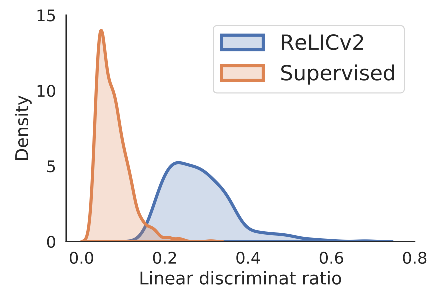

Figure 7. Distribution of the *linear discriminant ratio*: the ratio of between-class distances and within-class distances of embeddings computed on the ImageNet test set.

We see that there is a performance plateau going beyond 6 large views and a slight performance penalty going beyond 4 large views. For small views, we also observes performance penalties going beyond 2 small views, while there is a significant performance boost going from no small views to 2 small views, i.e. $+1.4\%$ and $+1.3\%$ in the case of 2 and 4 large views respectively. Note that this is double the performance improvement one gets from adding 2 large views, i.e the difference between $[2, 0]$ and $[4, 0]$ of $+0.7\%$. Furthermore, having 2 small views significantly reduces the generalization gap (difference between train and test error) compared to not having small views, i.e. we observe a relative decrease of $30+\%$. This supports our hypothesis that small views significantly contribute to learning more robust representations. Note that this is exactly the opposite as compared to (Caron et al., 2018) which argue for using smaller views as computationally less expensive alternatives to large views; in particular, they argue for using 2 large views and 6 small views.

G.5. Saliency masking

We measure the top-1 accuracy under linear evaluation on ImageNet. First, to isolate the contribution of saliency masking we measure the performance gain when applying saliency masking to just 2 large views; this improves performance from 74.8% to 75.3%, i.e. a gain of $+0.5\%$ which is a boost comparable to having two additional large views (see above). Next, we for different probabilities p_m of removing the background of the large augmented views during training.

p_m	0.0	0.1	0.15	0.2	0.25
Top-1	76.8	**77.1**	76.8	76.8	76.7

Applying the saliency masks 10% of the time results in the best performance and significantly improves over not using masking ($p_m = 0$). Moreover, we also explored using different datasets for pretraining our unsupervised saliency masking pipeline. We found that our pipeline is robust to the choice of pretraining dataset as varying this data had little effect on the results; see the appendix for details.

G.6. Invariance

To ascertain the importance of enforcing invariance over background removal and object styles, we compare ReLICv2 to SimCLR (Chen et al., 2020a) with different size views and saliency masking. We train both methods for 100 epochs and report top-1 test accuracy on ImageNet and use 4 large views and 2 small views. The SimCLR baseline (i.e. the standard setting without different views and saliency masking) is 64.5% while it achieves 66.2% when using different views and saliency masking, i.e. there is a gain of $+1.7\%$. On the other hand, without different size views and saliency masking, we achieve 61.1% while with saliency masking and different size views we get 67.5%, i.e. a gain of $+6.4\%$. Thus, invariance plays a crucial role in learning better representations.

H. Further ablations

In order to determine the sensitivity of ReLICv2 to different model hyperparameters, we perform an extensive ablation study. Unless otherwise noted, in this section we report results after 300 epochs of pretraining. As saliency masking is one of the main additions of ReLICv2 on top of ReLIC and was not covered extensively in the main text, we start our ablation analysis with looking into the effect of different modelling choices for it.

H.1. Using different datasets for obtaining the saliency masks

In the main text in Sections 3, 3, 3, 3 we used a saliency detection network trained only on a randomly selected subset of 2500 ImageNet images using the refinement mechanism proposed by DeepUSPS (Nguyen et al., 2019). Here we explore whether using additional data could help improve the performance of the saliency estimation and of the overall representations learnt by ReLICv2. For this purpose, we use the MSRA-B dataset (Liu et al., 2010), which was originally used by DeepUSPS to train their saliency detection network. MSRA-B consists of 2500 training images for which handcrafted masks computed with the methods Robust Background Detection (RBD) (Zhu et al., 2014), Hierarchy-associated Rich Features (HS) (Zou and Komodakis, 2015), Dense and Sparse Reconstruction (DSR) (Li et al., 2013) and Markov Chain (MC) (Jiang et al., 2013) are already available. We use the same hyperparameters as described in Section B.2.1 to train our saliency detection network on MSRA-B.

We explored whether using saliency masks obtained from training the saliency detection network on the MSRA-B affects performance of ReLICv2 pre-training on ImageNet. We noticed that for ReLICv2 representations pretrained on ImageNet for 1000 epochs, we get 77.2% top-1 and 93.3% top-5 accuracy under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder. The slight performance gains may due to the larger variety of images in MSRA-B used for training the saliency detection network, as opposed to the random sample of 2500 ImageNet images that we used for training the saliency detection network directly on the ImageNet dataset.

We also explored training the saliency detection network on 5000 randomly selected images from the ImageNet dataset and this resulted in the model overfitting, which degraded the quality of the saliency masks and resulted in a ReLICv2 performance of 76.7% top-1 and 93.3% top-5 accuracy on the ImageNet test set after 1000 epochs of pretraining on ImageNet training set.

The results for ReLICv2 in Section E are obtained by applying the saliency detection network trained on MSRA-B to all images in JFT-300M and then applying the saliency masks to the large augmented views during training as described in Section B.2.

H.2. Analysis and ablations for saliency masks

Using saliency masking during ReLICv2 training enables us to learn representations that focus on the semantically-relevant parts of the image, i.e. the foreground objects, and as such the learned representations should be more robust to background changes. We investigate the impact of using saliency masks with competing self-supervised benchmarks, the effect of the probability p_m of applying the saliency mask to each large augmented view during training as well as the robustness of ReLICv2 to random masks and mask corruptions. For the ablation experiments described in this section, we train the models for 300 epochs.

Using saliency masks with competing self-supervied methods. We evaluate the impact of using saliency masks with competing self-supervised methods such as BYOL (Grill et al., 2020). This method only uses two large augmentented views during training and we randomly apply the saliency masks, in a similar way as described in Section B.2, to each large augmented view with probability p_m. We report in Table 14 the top-1 and top-5 accuracy under linear evaluation on ImageNet for different settings of p_m for removing the background of the augmented images. We notice that saliency masking also helps to improve performance of BYOL.

Mask apply probability. We also investigate the effect of using probabilities ranging from 0 to 1 for applying the saliency mask during training for ReLICv2. In addition, we explore further the effect of using different datasets for training the saliency detection network that is subsequently used for computing the saliency masks. Table 15 reports the top-1 and top-5 accuracy for varying the mask apply probability p_m between 0 and 1 and for using the ImageNet vs. the MSRA-B dataset (Liu et al., 2010) for training our saliency detection network. Note that using the additional images from the MSRA-B

	Mask probability p_m	0	0.1	0.15	0.2	0.25	0.3
BYOL	Top-1	73.1	**73.4**	73.2	73.3	72.8	71.8
	Top-5	91.2	**91.3**	91.2	**91.3**	90.8	90.1

Table 14. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for BYOL trained using different probabilities of using the saliency mask to remove the background of the augmented images. Models are trained for 300 epochs.

dataset to train the saliency detection network results in better saliency masks which translates to better performance when using the saliency masks during RELICv2 training.

	Saliency network trained on ImageNet		Saliency network trained on MSRA-B	
Mask probability p_m	Top-1	Top-5	Top-1	Top-5
0	75.2	92.4	75.2	92.4
0.05	75.3	**92.6**	75.2	**92.6**
0.1	**75.4**	92.5	75.3	92.4
0.15	75.2	92.5	75.5	92.5
0.2	75.2	92.5	**75.6**	**92.6**
0.25	75.0	92.3	75.3	92.5
0.3	75.1	92.3	74.8	92.4
0.4	75.0	92.3	75.3	92.5
0.5	74.7	92.2	75.0	92.4
0.6	75.0	92.3	75.0	92.3
0.7	74.4	92.3	74.6	92.0
0.8	73.9	91.7	75.0	92.1
0.9	74.0	91.7	74.6	92.0
1.0	73.7	91.7	74.5	92.0

Table 15. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder set for different probabilities p_m of using the saliency mask to remove the background of the large augmented views during training and for using different datasets to train the saliency detection network for computing the saliency masks. Models are trained for 300 epochs.

Random masks and mask corruptions. To understand how important having accurate saliency masks for the downstream performance of representations is we also investigated using random masks, corrupting the saliency masks obtained from our saliency detection network and using a bounding box around the saliency masks during RELICv2 training.

We explored using completely random masks, setting the saliency mask to be a random rectangle of the image and also a centered rectangle. As ImageNet images generally consists of images with objects centered in the middle of the image, we expect that using a random rectangle that is centered around the middle will cover a reasonable portion of the object. Table 16 reports the performance under linear evaluation on the ImageNet test set when varying the size of the random masks to cover different percentage areas a_p of the full image. We notice that improving the quality of the masks, by using random rectangle patches instead of completely random points in the image as the mask, results in better performance. However, the performance with random masks is $> 1\%$ lower than using saliency masks from our saliency detection network. As expected, using centered rectangles instead of randomly positioned rectangles as masks results in better peformance.

Moreover, to test the robustness of RELICv2 to corruptions of the saliency masks, we add/remove from the masks a rectangle proportional to the area of the saliency mask. The mask rectangle is added/removed from the image center. Table 17 reports the results when varying the area of the rectangle to be added/removed to cover different percentages m_p of the saliency masks. We notice that while RELICv2 is robust to small corruptions of the saliency mask its performance drops in line with the quality of the saliency masks degrading.

Finally, we also explore corrupting the masks using a bounding box around the saliency mask which results in 74.5% top-1 and 92.2% top-5 accuracy under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder trained for 300 epochs with mask apply probability of 0.1 Note that this performance is comparable to using random rectangles to mask the large augmented views during training (see Table 16) and is lower than directly using the saliency masks from the trained

Image percentage area a_p	Random		Rectangle		Centered Rectangle	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
10%	70.8	89.9	70.9	90.3	71.3	90.1
20%	72.2	90.7	73.1	91.3	73.4	91.3
30%	72.9	91.3	73.8	91.8	73.8	91.9
40%	73.1	91.4	74.2	91.9	74.1	92.0
50%	73.3	91.5	74.0	92.0	74.3	92.0
60%	73.6	91.8	74.2	92.1	74.3	**92.2**
70%	73.7	91.9	**74.4**	92.1	**74.4**	**92.2**
80%	**74.1**	92.1	**74.4**	**92.2**	74.2	92.1
90%	**74.1**	**92.2**	**74.4**	92.1	74.2	**92.2**

Table 16. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder set for using different types of random masks that cover various percentage areas (a_p) of the full image. These random masks are applied on top of the large augmented views during training with probability 0.1. Models are trained for 300 epochs.

Mask percentage area m_p	Add rectangle to mask		Remove rectangle from mask	
	Top-1	Top-5	Top-1	Top-5
10%	75.2	92.5	**75.2**	92.3
20%	**75.3**	**92.6**	75.1	**92.4**
30%	75.1	92.3	74.7	92.2
40%	74.9	92.2	74.6	92.2
50%	74.9	92.4	74.5	92.0
60%	74.9	92.2	74.0	91.7
70%	74.8	92.2	73.6	91.7
80%	74.8	92.4	73.4	91.4
90%	74.7	92.2	73.0	91.3
100%	74.6	92.3	72.6	90.9

Table 17. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder set for corrupting the saliency masks by adding/remove a rectangle from the image center. The rectangle is a percentage (m_p) of the saliency mask area (the higher the percentage the higher the corruption). The corrupted saliency masks are applied on top of the large augmented views during training with probability 0.1.

saliency detection network.

H.3. Other model hyperparameters

Now we turn our attention to ablating the effect of other model hyperparameters on the downstream performance of RELICv2 representations. Note that these hyperparameters have been introduced and extensively ablated in prior work (Grill et al., 2020; Mitrovic et al., 2021; 2020).

Number of negatives. As mentioned in Section 2 RELICv2 selects negatives by randomly subsampling the minibatch in order to avoid false negatives. We investigate the effect of changing number of negatives in Table 18. We can see that the best performance can be achieved with relatively low numbers of negatives, i.e. just 10 negatives. Furthermore, we see that using the whole batch as negatives has one of the lowest performances.

In further experiments, we observed that for longer pretraining (e.g. 1000 epochs) there is less variation in performance than for pretraining for 300 epoch which itself is also quite low.

Target EMA. RELICv2 uses a target network whose weights are an exponential moving average (EMA) of the online encoder network which is trained normally using stochastic gradient descent; this is a setup first introduced in (Grill et al., 2020) and subsequently used in (Mitrovic et al., 2021) among others. The target network weights at iteration t are $\xi_t = \gamma\xi_{t-1} + (1-\gamma)\theta_t$ where γ is the EMA parameter which controls the stability of the target network ($\gamma = 0$ sets

Number of negatives	Top-1	Top-5
1	75.1	92.4
5	75.2	92.6
10	75.4	92.5
20	75.3	**92.7**
50	**75.5**	92.5
100	75.4	92.5
500	75.1	92.4
1000	75.3	92.6
2000	75.4	92.5
4096	75.2	92.6

Table 18. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder set for different numbers of randomly selected negatives. All settings are trained for 300 epochs.

$\xi_t = \theta_t$); θ_t are the parameters of the online encoder at time t, while ξ_t are the parameters of the target encoder at time t. As can be seen from Table 19, all decay rates between 0.9 and 0.996 yield similar performance for top-1 accuracy on the ImageNet test set after pretraining for 300 epochs indicating that ReLICv2 is robust to choice of γ in that range. For values of γ of 0.999 and higher, the performance quickly degrades indicating that the updating of the target network is too slow. Note that contrary to (Grill et al., 2020) where top-1 accuracy drops below 20% for $\gamma = 1$, ReLICv2 is significantly more robust to this setting achieving double that accuracy.

γ	Top-1	Top-5
0	73.5	91.5
0.9	74.6	92.2
0.99	**75.5**	**92.6**
0.993	75.4	92.5
0.996	74.4	92.0
0.999	70.5	89.8
1.0	39.6	63.6

Table 19. Top-1 and top-5 accuracy (in %) under linear evaluation on the ImageNet test set for a ResNet50 (1x) encoder set for different setting of the target exponentially moving average (EMA). All settings are trained for 300 epochs.