Barlow Twins: Self-Supervised Learning via Redundancy Reduction

Jure Zbontar [*1] Li Jing [*1] Ishan Misra [1] Yann LeCun [1 2] Stéphane Deny [1]

Abstract

Self-supervised learning (SSL) is rapidly closing the gap with supervised methods on large computer vision benchmarks. A successful approach to SSL is to learn embeddings which are invariant to distortions of the input sample. However, a recurring issue with this approach is the existence of trivial constant solutions. Most current methods avoid such solutions by careful implementation details. We propose an objective function that naturally avoids collapse by measuring the cross-correlation matrix between the outputs of two identical networks fed with distorted versions of a sample, and making it as close to the identity matrix as possible. This causes the embedding vectors of distorted versions of a sample to be similar, while minimizing the redundancy between the components of these vectors. The method is called BARLOW TWINS, owing to neuroscientist H. Barlow's *redundancy-reduction principle* applied to a pair of identical networks. BARLOW TWINS does not require large batches nor asymmetry between the network twins such as a predictor network, gradient stopping, or a moving average on the weight updates. Intriguingly it benefits from very high-dimensional output vectors. BARLOW TWINS outperforms previous methods on ImageNet for semi-supervised classification in the low-data regime, and is on par with current state of the art for ImageNet classification with a linear classifier head, and for transfer tasks of classification and object detection.[1]

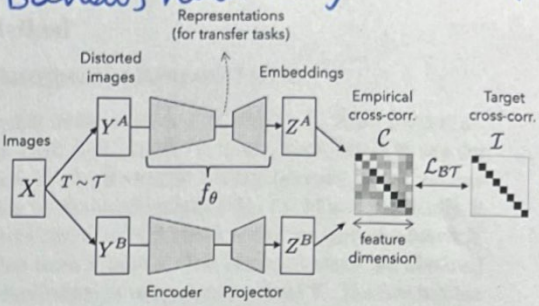

Figure 1. BARLOW TWINS's objective function measures the cross-correlation matrix between the embeddings of two identical networks fed with distorted versions of a batch of samples, and tries to make this matrix close to the identity. This causes the embedding vectors of distorted versions of a sample to be similar, while minimizing the redundancy between the components of these vectors. BARLOW TWINS is competitive with state-of-the-art methods for self-supervised learning while being conceptually simpler, naturally avoiding trivial constant (i.e. collapsed) embeddings, and being robust to the training batch size.

1. Introduction

Self-supervised learning aims to learn useful representations of the input data without relying on human annotations. Recent advances in self-supervised learning for visual data (Caron et al., 2020; Chen et al., 2020a; Grill et al., 2020; He et al., 2019; Misra & van der Maaten, 2019) show that it is possible to learn self-supervised representations that are competitive with supervised representations. A common underlying theme that unites these methods is that they all aim to learn representations that are invariant under different distortions (also referred to as 'data augmentations'). This

[*]Equal contribution [1]Facebook AI Research [2]New York University, NY, USA. Correspondence to: Jure Zbontar <jzb@fb.com>, Li Jing <ljng@fb.com>, Ishan Misra <imisra@fb.com>, Yann LeCun <yann@fb.com>, Stéphane Deny <stephane.deny.pro@gmail.com>.

Proceedings of the 38[th] International Conference on Machine Learning, PMLR 139, 2021.

[1]Code and pre-trained models (in PyTorch) are available at https://github.com/facebookresearch/barlowtwins

is typically achieved by maximizing similarity of representations obtained from different distorted versions of a sample using a variant of Siamese networks (Hadsell et al., 2006). As there are trivial solutions to this problem, like a constant representation, these methods rely on different mechanisms to learn useful representations.

Contrastive methods like SIMCLR (Chen et al., 2020a) define 'positive' and 'negative' sample pairs which are treated differently in the loss function. Additionally, they can also use asymmetric learning updates wherein momentum encoders (He et al., 2019) are updated separately from the main network. Clustering methods use one distorted sample to compute 'targets' for the loss, and another distorted version of the sample to predict these targets, followed by an alternate optimization scheme like k-means in DEEPCLUSTER (Caron et al., 2018) or non-differentiable operators in SWAV (Caron et al., 2020) and SELA (Asano et al., 2020). In another recent line of work, BYOL (Grill et al., 2020) and SIMSIAM (Chen & He, 2020), both the network architecture and parameter updates are modified to introduce asymmetry. The network architecture is modified to be asymmetric using a special 'predictor' network and the parameter updates are asymmetric such that the model parameters are only updated using one distorted version of the input, while the representations from another distorted version are used as a fixed target. (Chen & He, 2020) conclude that the asymmetry of the learning update, 'stop-gradient', is critical to preventing trivial solutions.

In this paper, we propose a new method, BARLOW TWINS, which applies *redundancy-reduction* — a principle first proposed in neuroscience — to self-supervised learning. In his influential article *Possible Principles Underlying the Transformation of Sensory Messages* (Barlow, 1961), neuroscientist H. Barlow hypothesized that the goal of sensory processing is to recode highly redundant sensory inputs into a factorial code (a code with statistically independent components). This principle has been fruitful in explaining the organization of the visual system, from the retina to cortical areas (see (Barlow, 2001) for a review and (Lindsey et al., 2020; Ocko et al., 2018; Schwartz & Simoncelli, 2001) for recent efforts), and has led to a number of algorithms for supervised and unsupervised learning (Ballé et al., 2017; Deco & Parra, 1997; Földiák, 1990; Linsker, 1988; Redlich, 1993a;b; Schmidhuber et al., 1996). Based on this principle, we propose an objective function which tries to make the cross-correlation matrix computed from twin embeddings as close to the identity matrix as possible. BARLOW TWINS is conceptually simple, easy to implement and learns useful representations as opposed to trivial solutions. Compared to other methods, it does not require large batches (Chen et al., 2020a), nor does it require any asymmetric mechanisms like prediction networks (Grill et al., 2020), momentum encoders (He et al., 2019), non-differentiable operators (Caron et al., 2020) or stop-gradients (Chen & He, 2020). Intriguingly, BARLOW TWINS strongly benefits from the use of very high-dimensional embeddings. BARLOW TWINS outperforms previous methods on ImageNet for semi-supervised classification in the low-data regime (55% top-1 accuracy for 1% labels), and is on par with current state of the art for ImageNet classification with a linear classifier head, as well as for a number of transfer tasks of classification and object detection.

2. Method

2.1. Description of BARLOW TWINS

Like other methods for SSL (Caron et al., 2020; Chen et al., 2020a; Grill et al., 2020; He et al., 2019; Misra & van der Maaten, 2019), BARLOW TWINS operates on a joint embedding of distorted images (Fig. 1). More specifically, it produces two distorted views for all images of a batch X sampled from a dataset. The distorted views are obtained via a distribution of data augmentations $\mathcal{T}$. The two batches of distorted views Y^A and Y^B are then fed to a function f_θ, typically a deep network with trainable parameters θ, producing batches of embeddings Z^A and Z^B respectively. To simplify notations, Z^A and Z^B are assumed to be mean-centered along the batch dimension, such that each unit has mean output 0 over the batch.

BARLOW TWINS distinguishes itself from other methods by its innovative loss function $\mathcal{L}_{\mathcal{BT}}$:

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \quad (1)$$

where λ is a positive constant trading off the importance of the first and second terms of the loss, and where $\mathcal{C}$ is the cross-correlation matrix computed between the outputs of the two identical networks along the batch dimension:

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}} \quad (2)$$

where b indexes batch samples and i, j index the vector dimension of the networks' outputs. $\mathcal{C}$ is a square matrix with size the dimensionality of the network's output, and with values comprised between -1 (i.e. perfect anti-correlation) and 1 (i.e. perfect correlation).

Intuitively, the *invariance term* of the objective, by trying to equate the diagonal elements of the cross-correlation matrix to 1, makes the embedding invariant to the distortions applied. The *redundancy reduction term*, by trying to equate the off-diagonal elements of the cross-correlation

matrix to 0, decorrelates the different vector components of the embedding. This decorrelation reduces the redundancy between output units, so that the output units contain non-redundant information about the sample.

More formally, BARLOW TWINS's objective function can be understood through the lens of information theory, and specifically as an instanciation of the *Information Bottleneck (IB)* objective (Tishby & Zaslavsky, 2015; Tishby et al., 2000). Applied to self-supervised learning, the IB objective consists in finding a representation that conserves as much information about the sample as possible while being the *least* possible informative about the specific distortions applied to that sample. The mathematical connection between BARLOW TWINS's objective function and the IB principle is explored in Appendix A.

BARLOW TWINS' objective function has similarities with existing objective functions for SSL. For example, the redundancy reduction term plays a role similar to the *contrastive term* in the INFONCE objective (Oord et al., 2018), as discussed in detail in Section 5. However, important conceptual differences in these objective functions result in practical advantages of our method compared to INFONCE-based methods, namely that (1) our method does not require a large number of negative samples and can thus operate on small batches (2) our method benefits from very high-dimensional embeddings. Alternatively, the redundancy reduction term can be viewed as a *soft-whitening* constraint on the embeddings, connecting our method to a recently proposed method performing a *hard-whitening* operation on the embeddings (Ermolov et al., 2020), as discussed in Section 5. However, our method performs better than current hard-whitening methods.

The pseudocode for BARLOW TWINS is shown as Algorithm 1.

Algorithm 1 PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the embeddings
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute embeddings
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

2.2. Implementation Details

Image augmentations Each input image is transformed twice to produce the two distorted views shown in Figure 1. The image augmentation pipeline consists of the following transformations: random cropping, resizing to 224×224, horizontal flipping, color jittering, converting to grayscale, Gaussian blurring, and solarization. The first two transformations (cropping and resizing) are always applied, while the last five are applied randomly, with some probability. This probability is different for the two distorted views in the last two transformations (blurring and solarization). We use the same augmentation parameters as BYOL (Grill et al., 2020).

Architecture The encoder consists of a ResNet-50 network (He et al., 2016) (without the final classification layer, 2048 output units) followed by a projector network. The projector network has three linear layers, each with 8192 output units. The first two layers of the projector are followed by a batch normalization layer and rectified linear units. We call the output of the encoder the 'representations' and the output of the projector the 'embeddings'. The representations are used for downstream tasks and the embeddings are fed to the loss function of BARLOW TWINS.

Optimization We follow the optimization protocol described in BYOL (Grill et al., 2020). We use the LARS optimizer (You et al., 2017) and train for 1000 epochs with a batch size of 2048. We however emphasize that our model works well with batches as small as 256 (see Ablations). We use a learning rate of 0.2 for the weights and 0.0048 for the biases and batch normalization parameters. We multiply the learning rate by the batch size and divide it by 256. We use a learning rate warm-up period of 10 epochs, after which we reduce the learning rate by a factor of 1000 using a cosine decay schedule (Loshchilov & Hutter, 2016). We ran a search for the trade-off parameter λ of the loss function and found the best results for $\lambda = 5 \cdot 10^{-3}$. We use a weight decay parameter of $1.5 \cdot 10^{-6}$. The biases and batch normalization parameters are excluded from LARS adaptation and weight decay. Training is distributed across 32 V100 GPUs and takes approximately 124 hours. For comparison, our reimplementation of BYOL trained with a batch size of 4096 takes 113 hours on the same hardware.

3. Results

We follow standard practice (Goyal et al., 2019) and evaluate our representations by transfer learning to different datasets and tasks in computer vision. Our network is pretrained using self-supervised learning on the training set of the ImageNet ILSVRC-2012 dataset (Deng et al., 2009) (without labels). We evaluate our model on a variety of tasks such as image classification and object detection, and using fixed representations from the network or finetuning it. We provide the hyperparameters for all the transfer learning experiments in the Appendix.

3.1. Linear and Semi-Supervised Evaluations on ImageNet

Linear evaluation on ImageNet We train a linear classifier on ImageNet on top of fixed representations of a ResNet-50 pretrained with our method. The top-1 and top-5 accuracies obtained on the ImageNet validation set are reported in Table 1. Our method obtains a top-1 accuracy of 73.2% which is comparable to the state-of-the-art methods.

Table 1. **Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet**. All models use a ResNet-50 encoder. Top-3 best self-supervised methods are underlined.

Method	Top-1	Top-5
Supervised	76.5	
MoCo	60.6	
PIRL	63.6	-
SimCLR	69.3	89.0
MoCo v2	71.1	90.1
SimSiam	71.3	-
SwAV (w/o multi-crop)	71.8	-
BYOL	74.3	91.6
SwAV	75.3	-
Barlow Twins (ours)	73.2	91.0

Semi-supervised training on ImageNet We fine-tune a ResNet-50 pretrained with our method on a subset of ImageNet. We use subsets of size 1% and 10% using the same split as SimCLR. The semi-supervised results obtained on the ImageNet validation set are reported in Table 2. Our method is either on par (when using 10% of the data) or slightly better (when using 1% of the data) than competing methods.

Table 2. **Semi-supervised learning on ImageNet** using 1% and 10% training examples. Results for the supervised method are from (Zhai et al., 2019). Best results are in **bold**.

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised	25.4	56.4	48.4	80.4
PIRL	-	-	57.2	83.8
SimCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SwAV	53.9	**70.2**	78.5	**89.9**
Barlow Twins (ours)	**55.0**	69.7	**79.2**	89.3

Table 3. **Transfer learning: image classification.** We benchmark learned representations on the image classification task by training linear classifiers on fixed features. We report top-1 accuracy on Places-205 and iNat18 datasets, and classification mAP on VOC07. Top-3 best self-supervised methods are underlined.

Method	Places-205	VOC07	iNat18
Supervised	53.2	87.5	46.7
SimCLR	52.5	85.5	37.2
MoCo-v2	51.8	86.4	38.6
SwAV (w/o multi-crop)	52.8	86.4	39.5
SwAV	56.7	88.9	48.6
BYOL	54.0	86.6	47.6
Barlow Twins (ours)	54.1	86.2	46.5

3.2. Transfer to other datasets and tasks

Image classification with fixed features We follow the setup from (Misra & van der Maaten, 2019) and train a linear classifier on fixed image representations, *i.e.*, the parameters of the ConvNet remain unchanged. We use a diverse set of datasets for this evaluation - Places-205 (Zhou et al., 2014) for scene classification, VOC07 (Everingham et al., 2010) for multi-label image classification, and iNaturalist2018 (Van Horn et al., 2018) for fine-grained image classification. We report our results in Table 3. Barlow Twins performs competitively against prior work, and outperforms SimCLR and MoCo-v2 on most datasets.

Object Detection and Instance Segmentation We evaluate our representations for the localization based tasks of object detection and instance segmentation. We use the VOC07+12 (Everingham et al., 2010) and COCO (Lin et al., 2014) datasets following the setup in (He et al., 2019) which finetunes the ConvNet parameters. Our results in Table 4 indicate that Barlow Twins performs comparably or better than state-of-the-art representation learning methods for these localization tasks.

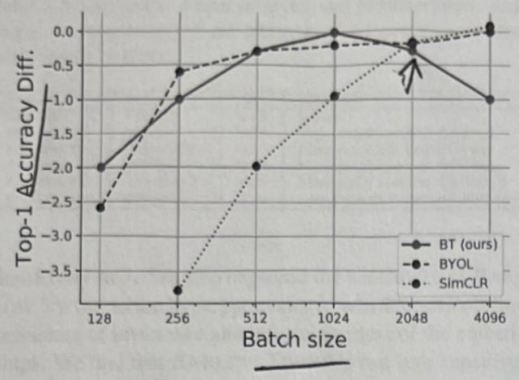

Figure 2. Effect of batch size. To compare the effect of the batch size across methods, for each method we report the difference between the top-1 accuracy at a given batch size and the best obtained accuracy among all batch size tested. BYOL: best accuracy is 72.5% for a batch size of 4096 (data from (Grill et al., 2020) fig. 3A). SIMCLR: best accuracy is 67.1% for a batch size of 4096 (data from (Chen et al., 2020a) fig. 9, model trained for 300 epochs). BARLOW TWINS: best accuracy is 71.7% for a batch size of 1024.

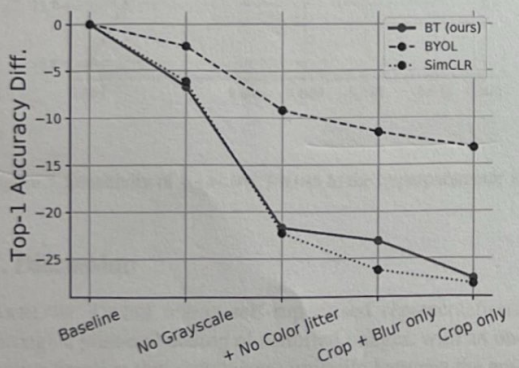

Figure 3. Effect of progressively removing data augmentations. Data for BYOL and SIMCLR (repro) is from (Grill et al., 2020) fig 3b.

In stark contrast, we find that BARLOW TWINS performs better when the dimensionality of the projector network output is very large. Other methods rapidly saturate when the dimensionality of the output increases, but our method keeps improving with all output dimensionality tested (Fig. 4). This result is quite surprising because the output of the ResNet is kept fixed to 2048, which acts as a dimensionality bottleneck in our model and sets the limit of the intrinsic dimensionality of the representation. In addition, similarly to other methods, we find that our model performs better when the projector network has more layers, with a saturation of the performance for 3 layers.

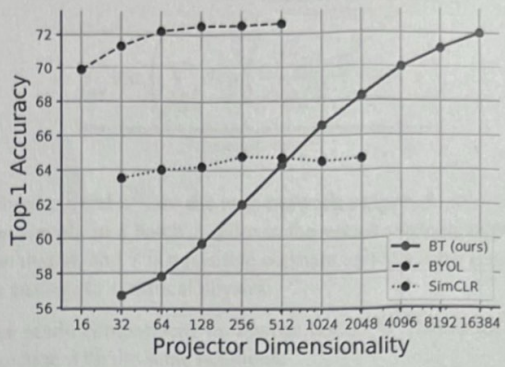

Figure 4. Effect of the dimensionality of the last layer of the projector network on performance. The parameter λ is kept fix for all dimensionalities tested. Data for SIMCLR is from (Chen et al., 2020a) fig 8; Data for BYOL is from (Grill et al., 2020) Table 14b.

Breaking Symmetry Many SSL methods (e.g. BYOL, SIMSIAM, SWAV) rely on different symmetry-breaking mechanisms to avoid trivial solutions. Our loss function avoids these trivial solutions by construction, even in the case of symmetric networks. It is however interesting to ask whether breaking symmetry can further improve the performance of our network. Following SIMSIAM and BYOL, we experiment with adding a predictor network composed of 2 fully connected layers of size 8192 to one of the network (with batch normalization followed by a ReLU nonlinearity in the hidden layer) and/or a stop-gradient mechanism on the other network. We find that these asymmetries slightly decrease the performance of our network (see Table 6).

Table 6. Effect of asymmetric settings

case	stop-gradient	predictor	Top-1	Top-5
Baseline	-	-	71.4	90.2
(a)	✓	-	70.5	89.0
(b)	-	✓	70.2	89.0
(c)	✓	✓	61.3	83.5

BYOL with a larger projector/predictor/embedding For a fair comparison with BYOL, we also evaluated BYOL with a wider and/or deeper projector head (3-layer MLP), a wider and/or deeper predictor head, and a larger dimensionality of the embedding. BYOL did not improve under these conditions (see Table 7).

Table 7. Wider and/or deeper projector and predictor heads and larger dimensionality of the embedding did not improve the performance of BYOL.

Projector	Predictor	Acc1	Description
4096-256	4096-256	74.1%	baseline
4096-4096-256	4096-256	74.0%	3 layer proj, 2 layer pred, 256-d repr.
4096-4096-256	4096-4096-256	73.2%	3 layer proj, 3 layer pred, 256-d repr.
4096-4096-512	4096-512	73.7%	3 layer proj, 2 layer pred, 512-d repr.
4096-4096-512	4096-4096-512	73.2%	3 layer proj, 3 layer pred, 512-d repr.
8192-8192-8192	8192-8192	72.3%	same proj as BT, 2 layer pred, 8192-d repr.

Sensitivity to λ. We also explored the sensitivity of BARLOW TWINS to the hyperparameter λ, which trades off the desiderata of invariance and informativeness of the embeddings. We find that BARLOW TWINS is not very sensitive to this hyperparameter (Fig. 5).

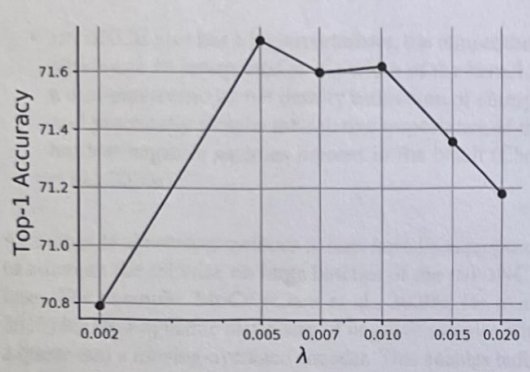

Figure 5. Sensitivity of BARLOW TWINS to the hyperparameter λ

5. Discussion

BARLOW TWINS learns self-supervised representations through a joint embedding of distorted images, with an objective function that maximizes similarity between the embedding vectors while reducing redundancy between their components. Our method does not require large batches of samples, nor does it require any particular asymmetry in the twin network structure. We discuss next the similarities and differences between our method and prior art, both from a conceptual and an empirical standpoint. For ease of comparison, all objective functions are recast with a common set of notations. The discussion ends with future directions.

5.1. Comparison with Prior Art

infoNCE The INFONCE loss, where NCE stands for Noise-Contrastive Estimation (Gutmann & Hyvärinen, 2010), is a popular type of contrastive loss function used for self-supervised learning (e.g. (Chen et al., 2020a; He et al., 2019; Hénaff et al., 2019; Oord et al., 2018)). It can be instantiated as:

$$\mathcal{L}_{infoNCE} \triangleq \underbrace{-\sum_b \frac{\langle z_b^A, z_b^B \rangle_i}{\tau \left\| z_b^A \right\|_2 \left\| z_b^B \right\|_2}}_{\text{similarity term}} + \underbrace{\sum_b \log \left(\sum_{b' \neq b} \exp \left(\frac{\langle z_b^A, z_{b'}^B \rangle_i}{\tau \left\| z_b^A \right\|_2 \left\| z_{b'}^B \right\|_2} \right) \right)}_{\text{contrastive term}}$$

where z^A and z^B are the twin network outputs, b indexes the sample in a batch, i indexes the vector component of the output, and τ is a positive constant called temperature in analogy to statistical physics.

For ready comparison, we rewrite BARLOW TWINS loss function with the same notations:

$$\mathcal{L}_{BT} = \underbrace{\sum_i \left(1 - \frac{\langle z_{\cdot,i}^A, z_{\cdot,i}^B \rangle_b}{\left\| z_{\cdot,i}^A \right\|_2 \left\| z_{\cdot,i}^B \right\|_2} \right)^2}_{\text{invariance term}} + \underbrace{\lambda \sum_i \sum_{j \neq i} \left(\frac{\langle z_{\cdot,i}^A, z_{\cdot,j}^B \rangle_b}{\left\| z_{\cdot,i}^A \right\|_2 \left\| z_{\cdot,j}^B \right\|_2} \right)^2}_{\text{redundancy reduction term}}$$

Both BARLOW TWINS' and INFONCE's objective functions have two terms, the first aiming at making the embeddings invariant to the distortions fed to the twin networks, the second aiming at maximizing the variability of the embedding learned. Another common point between the two losses is that they both rely on batch statistics to measure this variability. However, the INFONCE objective maximizes the variability of the embeddings by maximizing the pairwise distance between all pairs of samples, whereas our method does so by decorrelating the components of the embeddings vectors.

The contrastive term in INFONCE can be interpreted as a non-parametric estimation of the entropy of the distribution of embeddings (Wang & Isola, 2020). An issue that arises with non-parametric entropy estimators is that they are prone to the curse of dimensionality: they can only be estimated reliably in a low-dimensional setting, and they typically require a large number of samples.

In contrast, our loss can be interpreted as a *proxy* entropy estimator of the distribution of embeddings under *a Gaussian parametrization* (see Appendix A). Thanks to this simplified parametrization, the variability of the embedding can be estimated from much fewer samples, and on very large-dimensional embeddings. Indeed, in the ablation studies that we perform, we find that (1) our method is robust to small batches unlike the popular INFONCE-based method

SIMCLR, and (2) our method benefits from using very large dimensional embeddings, unlike INFONCE-based methods which do not see a benefit in increasing the dimensionality of the output.

Our loss presents several other interesting differences with infoNCE:

- In INFONCE, the embeddings are typically normalized along the feature dimension to compute a cosine similarity between embedded samples. We normalize the embeddings along the batch dimension instead.
- In our method, there is a parameter λ that trades off how much emphasis is put on the invariance term vs. the redundancy reduction term. This parameter can be interpreted as the trade-off parameter in the *Information Bottleneck* framework (see Appendix A). This parameter is not present in INFONCE.
- INFONCE also has a hyperparameter, the temperature, which can be interpreted as the width of the kernel in a non-parametric kernel density estimation of entropy, and practically weighs the relative importance of the hardest negative samples present in the batch (Chen et al., 2020a).

A number of alternative methods to ours have been proposed to alleviate the reliance on large batches of the INFONCE loss. For example, MoCo (Chen et al., 2020b; He et al., 2019) builds a dynamic dictionary of negative samples with a queue and a moving-averaged encoder. This enables building a large and consistent dictionary on-the-fly that facilitates contrastive unsupervised learning. MoCo typically needs to store $> 60{,}000$ sample embeddings. In contrast, our method does not require such a large dictionary, since it works well with a relatively small batch size (e.g. 256).

Asymmetric Twins BOOTSTRAP-YOUR-OWN-LATENT (aka BYOL) (Grill et al., 2020) and SIMSIAM (Chen & He, 2020) are two recent methods which use a simple cosine similarity between twin embeddings as an objective function, without *any* contrastive term:

$$\mathcal{L}_{cosine} = -\sum_b \frac{\langle z_b^A, z_b^B \rangle_i}{\left\| z_b^A \right\|_2 \left\| z_b^B \right\|_2}$$

Surprisingly, these methods successfully avoid trivial solutions by introducing some asymmetry in the architecture and learning procedure of the twin networks. For example, BYOL uses a predictor network which breaks the symmetry between the two networks, and also enforces an exponential moving average on the target network weights to slow down the progression of the weights on the target network. Combined together, these two mechanisms surprisingly avoid trivial solutions. The reasons behind this success are the subject of recent theoretical and empirical studies (Chen & He, 2020; Fetterman & Albrecht, 2020; Richemond et al., 2020; Tian et al., 2020). In particular, the ablation study (Chen & He, 2020) shows that the moving average is not necessary, but that stop-gradient on one of the branch and the presence of the predictor network are two crucial elements to avoid collapse. Other works show that batch normalization (Fetterman & Albrecht, 2020; Tian et al., 2020) or alternatively group normalization (Richemond et al., 2020) could play an important role in avoiding collapse.

Like our method, these asymmetric methods do not require large batches, since in their case there is no interaction between batch samples in the objective function.

It should be noted however that these asymmetric methods cannot be described as the optimization of an overall learning objective. Instead, there exists trivial solutions to the learning objective that these methods avoid via particular implementation choices and/or the result of non-trivial learning dynamics. In contrast, our method avoids trivial solutions by construction, making our method conceptually simpler and more principled than these alternatives (until their principle is discovered, see (Tian et al., 2021) for an early attempt).

Whitening In a concurrent work, (Ermolov et al., 2020) propose W-MSE. Acting on the embeddings from identical twin networks, this method performs a differentiable whitening operation (via Cholesky decomposition) of each batch of embeddings before computing a simple cosine similarity between the whitened embeddings of the twin networks. In contrast, the redundancy reduction term in our loss encourages the whitening of the batch embeddings as a soft constraint. The current W-MSE model achieves 66.3% top-1 accuracy on the Imagenet linear evaluation benchmark. It is an interesting direction for future studies to determine whether improved versions of this hard-whitening strategy could also lead to state-of-the-art results on these large-scale computer vision benchmarks.

Clustering These methods, such as DEEPCLUSTER (Caron et al., 2018), SWAV (Caron et al., 2020), SELA (Asano et al., 2020), perform contrastive-like comparisons without the requirement to compute all pairwise distances. Specifically, these methods simultaneously cluster the data while enforcing consistency between cluster assignments produced for different distortions of the same image, instead of comparing features directly as in contrastive learning. Clustering methods are also prone to collapse, *e.g.*, empty clusters in k-means and avoiding them relies on careful implementation details. Online clustering methods like SWAV

can be trained with large and small batches but require storing features when the number of clusters is much larger than the batch size. Clustering methods can also be combined with contrastive learning (Li et al., 2021) to prevent collapse.

Noise As Targets This method (Bojanowski & Joulin, 2017) learns to map samples to fixed random targets on the unit sphere, which can be interpreted as a form of whitening. This objective uses a single network, and hence does not leverage the distortions induced by twin networks. Predefining random targets might limit the flexibility of the representation that can be learned.

IMAX In the early days of SSL, (Becker & Hinton, 1992; Zemel & Hinton, 1990) proposed a loss function between twin networks given by:

$$\mathcal{L}_{IMAX} \triangleq \log|\mathcal{C}_{(Z^A - Z^B)}| - \log|\mathcal{C}_{(Z^A + Z^B)}|$$

where $|\,|$ denotes the determinant of a matrix, $\mathcal{C}_{(Z^A - Z^B)}$ is the covariance matrix of the difference of the outputs of the twin networks and $\mathcal{C}_{(Z^A + Z^B)}$ the covariance of the sum of these outputs. It can be shown that this objective maximizes the information between the twin network representations under the assumptions that the two representations are noisy versions of the same underlying Gaussian signal, and that the noise is independant, additive and Gaussian. This objective is similar to ours in the sense that there is one term that encourages the two representations to be similar and another term that encourages the units to be decorrelated. However, unlike IMAX, our objective is not directly an information quantity, and we have an extra trade-off parameter λ that trades off the two terms of our loss. The IMAX objective was used in early work so it is not clear whether it can scale to large computer vision tasks. Our attempts to make it work on ImageNet were not successful.

5.2. Future Directions

We observe a steady improvement of the performance of our method as we increase the dimensionality of the embeddings (i.e. of the last layer of the projector network). This intriguing result is in stark contrast with other popular methods for SSL, such as SIMCLR (Chen et al., 2020a) and BYOL (Grill et al., 2020), for which increasing the dimensionality of the embeddings rapidly saturates performance. It is a promising avenue to continue this exploration for even higher dimensional embeddings ($> 16{,}000$), but this would require the development of new methods or alternative hardware to accommodate the memory requirements of operating on such large embeddings.

Our method is just one possible instanciation of the *Information Bottleneck* principle applied to SSL. We believe that further refinements of the proposed loss function and algorithm could lead to more efficient solutions and even better performances. For example, the redundancy reduction term is currently computed from the off-diagonal terms of the cross-correlation matrix between the twin network embeddings, but alternatively it could be computed from the off-diagonal terms of the auto-correlation matrix of a single network's embedding. Our preliminary analyses seem to indicate that this alternative leads to similar performances (not shown). A modified loss could also be applied to the (unnormalized) cross-covariance matrix instead of the (normalized) cross-correlation matrix (see Ablations for preliminary analyses).

Acknowledgements

We thank Pascal Vincent, Yubei Chen and Samuel Ocko for helpful insights on the mathematical connection to the infoNCE loss, Robert Geirhos and Adrien Bardes for extra analyses not included in the manuscript and Xinlei Chen, Mathilde Caron, Armand Joulin, Reuben Feinman and Ulisse Ferrari for useful comments on the manuscript.

References

Asano, Y. M., Rupprecht, C., and Vedaldi, A. Self-labelling via simultaneous clustering and representation learning. *International Conference on Learning Representations (ICLR)*, 2020.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end Optimized Image Compression. In *International Conference on Learning Representations (ICLR)*, 2017.

Barlow, H. Redundancy reduction revisited. *Network (Bristol, England)*, 12(3):241–253, August 2001. ISSN 0954-898X.

Barlow, H. B. *Possible Principles Underlying the Transformations of Sensory Messages*. in Sensory Communication, The MIT Press, 1961.

Becker, S. and Hinton, G. E. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, January 1992.

Bojanowski, P. and Joulin, A. Unsupervised learning by predicting noise. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Cai, T. T., Liang, T., and Zhou, H. H. Law of Log Determinant of Sample Covariance Matrix and Optimal Estimation of Differential Entropy for High-Dimensional Gaussian Distributions. *Journal of Multivariate Analysis*, 137, 2015. doi: 10.1016/j.jmva.2015.02.003.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In