WHITENING FOR SELF-SUPERVISED REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most of the self-supervised learning methods are based on the *contrastive loss*, where image instances which share the same semantic content ("positives") are contrasted with instances extracted from other images ("negatives"). For the learning to be effective, a lot of negatives should be compared with a positive pair, which is computationally demanding. In this paper, we propose a different direction and a new loss function for self-supervised learning which is based on the *whitening* of the latent-space features. The whitening operation has a "scattering" effect on the batch samples, which compensates the use of negatives, avoiding degenerate solutions where all the sample representations collapse to a single point. Our Whitening MSE (W-MSE) loss does not require special heuristics (e.g. additional networks) and it is conceptually simple. Since negatives are not needed, we suggest obtaining multiple positive pairs from one image in the batch. We show empirically that W-MSE is competitive with respect to popular, more complex self-supervised methods. The source code of the method and all the experiments is included in the Supplementary Material.

1 INTRODUCTION

One of the current main bottlenecks in deep network training is the dependence on large annotated training datasets, and this motivates the recent surge of interest in unsupervised methods. Specifically, in self-supervised representation learning, a network is (pre-)trained without any form of manual annotation, thus providing a means to extract information from unlabeled-data sources (e.g., text corpora, videos, images from the Internet, etc.). In self-supervision, label information is replaced by a prediction problem using some form of *context* or using a *pretext* task. Pioneering work in this direction was done in Natural Language Processing (NLP), in which the co-occurrence of words in a sentence is used to learn a language model (Mikolov et al., 2013a;b; Devlin et al., 2019). In Computer Vision, typical contexts or pretext tasks are based on: (1) the temporal consistency in videos (Wang & Gupta, 2015; Misra et al., 2016; Dwibedi et al., 2019), (2) the spatial order of patches in still images (Noroozi & Favaro, 2016; Misra & van der Maaten, 2019; Hénaff et al., 2019) or (3) simple image transformation techniques (Ji et al., 2019; He et al., 2019; Wu et al., 2018). The intuitive idea behind most of these methods is to collect pairs of *positive* and *negative* samples: two positive samples should share the same semantics, while negatives should be perceptually different. A triplet loss (Sohn, 2016; Schroff et al., 2015; Hermans et al., 2017; Wang & Gupta, 2015; Misra et al., 2016) can then be used to learn a metric space which should represent the human perceptual similarity. However, most of the recent studies use a contrastive loss (Hadsell et al., 2006) or one of its variants (Gutmann & Hyvärinen, 2010; van den Oord et al., 2018; Hjelm et al., 2019), while Tschannen et al. (2019) show the relation between the triplet loss and the contrastive loss.

It is worth noticing that the success of both kinds of losses is strongly affected by the number and the quality of the negative samples. For instance, in the case of the triplet loss, a common practice is to select *hard/semi-hard* negatives (Schroff et al., 2015; Hermans et al., 2017). On the other hand, Hjelm et al. (2019) have shown that the contrastive loss needs a large number of negatives to be competitive. This implies using batches with a large size, which is computationally demanding, especially with high-resolution images. In order to alleviate this problem, Wu et al. (2018) use a *memory bank* of negatives, which is composed of feature-vector representations of all the training samples. He et al. (2019) conjecture that the use of large and fixed-representation vocabularies is one of the keys to the success of self-supervision in NLP. The solution proposed by He et al. (2019)

extends Wu et al. (2018) using a memory-efficient queue of the last visited negatives, together with a *momentum encoder* which preserves the intra-queue representation consistency. Chen et al. (2020) have performed large-scale experiments confirming that a large number of negatives (and therefore a large batch size) is required for the contrastive loss to be efficient. Concurrently with our work, Grill et al. (2020) have suggested that it is not necessary to rely on the contrastive scheme, introducing a high-performing alternative based on bootstrapping.

In this paper we propose a new self-supervised loss function which first *scatters* all the sample representations in a spherical distribution[1] and then *penalizes* the positive pairs which are far from each other. In more detail, given a set of samples $V = \{\mathbf{v}_i\}$, corresponding to the current mini-batch of images $B = \{x_i\}$, we first project the elements of V onto a spherical distribution using a *whitening* transform (Siarohin et al., 2019). The whitened representations $\{\mathbf{z}_i\}$, corresponding to V, are normalized and then used to compute a Mean Squared Error (MSE) loss which accumulates the error taking into account only positive pairs $(\mathbf{z}_i, \mathbf{z}_j)$. We do not need to *contrast* positives against negatives as in the contrastive loss or in the triplet loss because the optimization process leads to shrinking the distance between positive pairs and, indirectly, scatters the other samples to satisfy the overall spherical-distribution constraint.

In summary, our contributions are the following:

- We propose a new loss function Whitening MSE (W-MSE) for self-supervised training. W-MSE constrains the batch samples to lie in a spherical distribution and it is an alternative to positive-negative instance contrasting methods.
- Our loss does not rely on negatives, thus more positive samples in the batch can be beneficial; we demonstrate that multiple positive pairs produced from one image improve the performance.
- We empirically show that our W-MSE loss outperforms the commonly adopted contrastive loss when measured using different standard classification protocols. We show that W-MSE is competitive with respect to state-of-the-art self-supervised methods.

2 Background and Related Work

A typical self-supervised method is composed of two main components: a *pretext task*, which exploits some a-priori knowledge about the domain to automatically extract supervision from data, and a *loss function*. In this section we briefly review both aspects, and we additionally analyse the recent literature concerning feature whitening.

Pretext Tasks. The temporal consistency in a video provides an intuitive form of self-supervision: temporally-close frames usually contain a similar semantic content (Wang & Gupta, 2015; van den Oord et al., 2018). Misra et al. (2016) extended this idea using the relative temporal order of 3 frames, while Dwibedi et al. (2019) used a *temporal cycle consistency* for self-supervision, which is based on comparing two videos sharing the same semantics and computing inter-video frame-to-frame nearest neighbour assignments.

When dealing with still images, the most common pretext task is *instance discrimination*: an input image x_i is transformed into x_j using a (composition of) data-augmentation technique(s), such as image cropping, rotation, color jittering, Sobel filtering, etc., and then the learner is required to discriminate (x_i, x_j) from other samples (Wu et al., 2018; Ji et al., 2019; He et al., 2019; Chen et al., 2020).

Denoising auto-encoders (Vincent et al., 2008) add random noise to the input image and try to recover the original image. More sophisticated pretext tasks consist in predicting the spatial order of image patches (Noroozi & Favaro, 2016; Misra & van der Maaten, 2019) or in reconstructing large masked regions of the image (Pathak et al., 2016). Hjelm et al. (2019); Bachman et al. (2019) compare the holistic representation of an input image with a patch of the same image. Hénaff et al. (2019) use a similar idea, where the comparison depends on the patch order: the appearance of a given patch should be predicted given the appearance of the patches which lie above it in the image.

[1]Here and in the following, with "spherical distribution" we mean a distribution with a zero-mean and an identity-matrix covariance.

In this paper we use standard data augmentation techniques on still images to obtain positive pairs, which is a simple method to get self-supervision (Chen et al., 2020) and does not require a pretext-task specific network architecture (Hjelm et al., 2019; Bachman et al., 2019; Hénaff et al., 2019).

Loss functions. Denoising auto-encoders use a *reconstruction loss* which compares the generated image with the input image before adding noise. Other generative methods use an *adversarial loss* in which a discriminator provides supervisory information to the generator (Donahue et al., 2017; Donahue & Simonyan, 2019).

Early self-supervised (deep) discriminative methods used a *triplet loss* (Wang & Gupta, 2015; Misra et al., 2016): given two *positive* images x_i, x_j and a *negative* x_k (Sec. 1), together with their corresponding latent-space representations $\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k$, this loss penalizes those cases in which $\mathbf{z}_i$ and $\mathbf{z}_k$ are closer to each other than $\mathbf{z}_i$ and $\mathbf{z}_j$ plus a margin m:

$$L_{Triplet} = -\max(\mathbf{z}_i^T \mathbf{z}_k - \mathbf{z}_i^T \mathbf{z}_j + m, 0). \tag{1}$$

Most of the recent self-supervised discriminative methods are based on some *contrastive loss* (Hadsell et al., 2006) variant, in which $\mathbf{z}_i$ and $\mathbf{z}_j$ are contrasted against a set of negative pairs. Following the common formulation proposed by van den Oord et al. (2018):

$$L_{Contrastive} = -\log \frac{\exp\left(\mathbf{z}_i^T \mathbf{z}_j / \tau\right)}{\sum_{k=1, k\neq i}^{K} \exp\left(\mathbf{z}_i^T \mathbf{z}_k / \tau\right)}, \tag{2}$$

where τ is a *temperature* hyperparameter which should be manually set and the sum in the denominator is over a set of $K-1$ negative samples. Usually K is the size of the current batch, i.e., $K = 2N$, being N the number of the collected positive pairs. However, as shown by Hjelm et al. (2019), the contrastive loss (2) requires a large number of negative samples to be competitive. Wu et al. (2018); He et al. (2019) use a set of negatives much larger than the current batch, by pre-computing latent-space representations of old samples. Chen et al. (2020) use simple, but an exceedingly computationally demanding solution with large batches.

Wang & Isola (2020) have demonstrated that asymptotically the contrastive loss for normalized features can be decoupled into two components: one loss pushes the representation to preserve uniformity on the hypersphere, substituting negative samples, while second loss pulls positive samples closer to each other. Grill et al. (2020) have proposed an alternative to the positive-negative scheme based on bootstrapping. The main network is optimised to predict the output of the target network, which parameters are a running average of the main network. To avoid a degenerate constant solution the method includes an additional prediction network. Still, contrary to our proposal, this is only an empirical method, there is no theoretical guarantee that such solutions will be avoided.

In this paper we propose a different loss which is competitive with respect to other alternatives. Our loss formulation is simpler since it does not require a proper setting of the τ hyperparameter in equation 2, m in equation 1 or additional networks with a specific update scheme as Grill et al. (2020). Finally, while many recent works (van den Oord et al., 2018; Hénaff et al., 2019; Hjelm et al., 2019; Bachman et al., 2019; Ravanelli & Bengio, 2018) draw a relation between the contrastive loss and an estimate of the mutual information between latent-space image representations, Tschannen et al. (2019) argue that the success of this loss is likely related to learning a metric space, similarly to what happens with a triplet loss.

Feature Whitening. We adopt the efficient and stable Cholesky decomposition (Dereniowski & Marek, 2004) based *whitening* transform proposed by Siarohin et al. (2019) to project our latent-space vectors into a spherical distribution (more details in Sec. 3). Note that Huang et al. (2018); Siarohin et al. (2019) use whitening transforms in the intermediate layers of the network for a completely different task: extending Batch Normalization (Ioffe & Szegedy, 2015) to a multivariate batch normalization.

3 THE WHITENING MSE LOSS

Given an image x, we extract an embedding $\mathbf{z} = f(x; \theta)$ using a network $f(\cdot; \theta)$ parametrized with θ (more details below). We require that: (1) the image embeddings are drawn from a non-degenerate

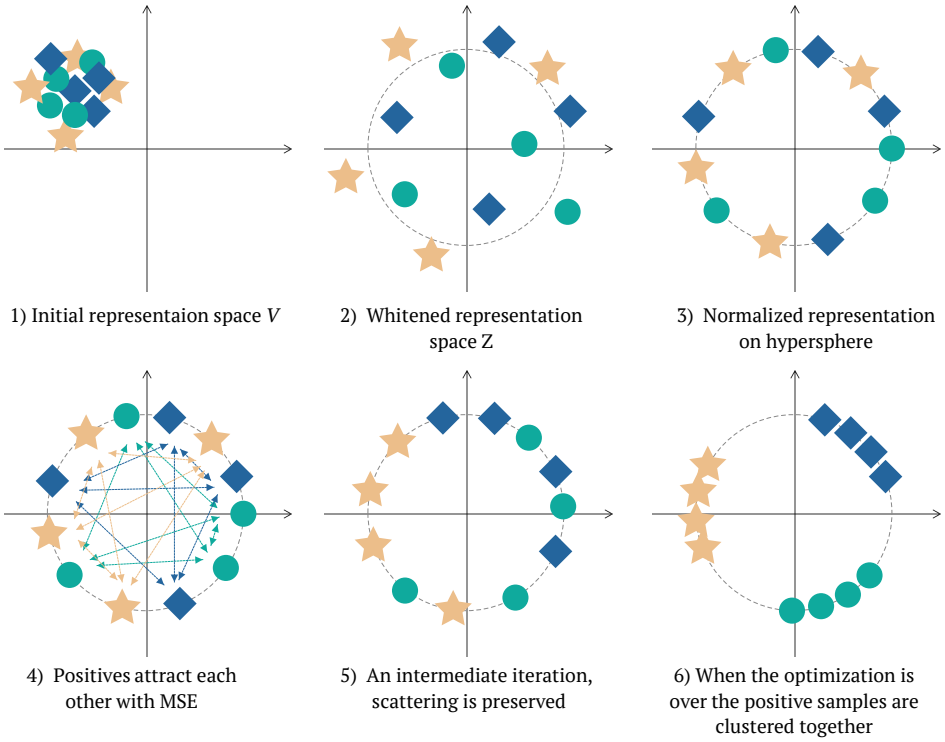

Figure 1: A schematic representation of the optimization process driven by our W-MSE loss. Positive pairs are indicated with the same shapes and colors. (1) A representation of the feature batch V when training starts. (2, 3) The distribution of the elements after whitening and normalization. (4) The MSE computed over normalized Z encourages the network to move the positive pair representations closer to each other. (5) The subsequent iterations move closer and closer the positive pairs, while the relative layout of the other samples is forced to lie in a spherical distribution.

distribution (the latter being a distribution where, e.g., all the representations collapse to a single point), and (2) positive image pairs (x_i, x_j), which share a similar semantics, should be clustered close to each other. We formulate this problem as follows:

$$min_\theta \, \mathbb{E} \, dist(\mathbf{z}_i, \mathbf{z}_j), \tag{3}$$

$$s.t. \; cov(\mathbf{z}_i, \mathbf{z}_i) = cov(\mathbf{z}_j, \mathbf{z}_j) = I, \tag{4}$$

where $dist$ is a distance between vectors, I is the identity matrix and $(\mathbf{z}_i, \mathbf{z}_j)$ correspond to a positive pair of images (x_i, x_j). With equation 4, we constrain the distribution of the $\mathbf{z}$ values to be non-degenerate, hence avoiding that all the probability mass is concentrated in a single point. Moreover, equation 4 makes all the components of $\mathbf{z}$ to be linearly independent from each other, which encourages the different dimensions of $\mathbf{z}$ to represent different semantic content. In most experiments we define the distance with cosine similarity, implemented with MSE between normalized vectors:

$$dist(\mathbf{z}_i, \mathbf{z}_j) = \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \right\|_2^2 = 2 - 2 \frac{\langle \mathbf{z}_i, \mathbf{z}_j \rangle}{\|\mathbf{z}_i\|_2 \cdot \|\mathbf{z}_j\|_2} \tag{5}$$

Additionally we include experiments using Euclidean distance (see Sec. 4.2). We provide below the details on how positive image samples are collected, how they are encoded and how equations are implemented.

First, similarly to Chen et al. (2020), we obtain positive samples sharing the same semantics from a single image x and using standard image transformation techniques. Specifically, we use a composition of image cropping, grayscaling and color jittering transformations $T(\cdot; \mathbf{p})$, whose parameters $(\mathbf{p})$ are selected uniformly at random and independently of each other in order to obtain *positive*

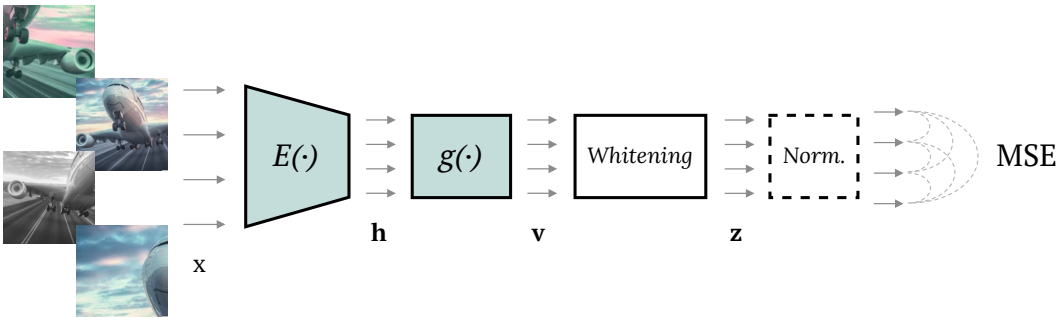

Figure 2: Scheme of the training procedure. First, 4 samples are generated using augmentations. These images are transformed into vectors with the encoder $E(\cdot)$. Next, they are projected to a lower dimensional space with a projection head $g(\cdot)$. Whitening transforms vectors to a spherical distribution, followed by optional normalization. Finally, dashed curves on the scheme show resulting 6 positive pairs.

samples from the same image: $x_i = T(x; \mathbf{p}_i)$. We concisely indicate with $pos(i, j)$ the fact that x_i and x_j ($x_i, x_j \in B$, B the current batch) are matched to each other because they share the same semantics.

The number of positive samples per image d may vary, trading off diversity in the batch and the amount of the training signal. Favoring more negatives, most of the methods use one positive pair ($d = 2$). However, Ji et al. (2019) have demonstrated improved performance with 5 samples, while Caron et al. (2020) use 8 samples. In our method, we use all possible combinations of the samples, thus the total number of positive pairs per image equals to $d(d-1)/2$. We include experiments for $d = 2$ (1 positive pair) and $d = 4$ (6 positive pairs).

For representation learning, we use a backbone *encoder* network $E(\cdot)$. $E(\cdot)$, trained without human supervision, will be used in Sec. 4 for evaluation using standard protocols. We use a standard ResNet-18 (He et al., 2016) as the encoder, and $\mathbf{h} = E(x)$ is the output of the average-pooling layer. This choice has the advantage to be simple and easily reproducible, in contrast to other methods who use encoder architectures specific for a given pretext task (see Sec. 2). Since $\mathbf{h} \in \mathbb{R}^{512}$ is a high-dimensional vector, following Chen et al. (2020) we use a nonlinear projection head $g(\cdot)$ to project $\mathbf{h}$ in a lower dimensional space: $\mathbf{v} = g(\mathbf{h})$, where $g(\cdot)$ is implemented with a MLP with one hidden layer and a batch normalization. The whole network $f(\cdot)$ is given by the composition of $g(\cdot)$ with $E(\cdot)$. The training procedure is depicted at Fig. 2.

Given N original images and a batch of samples $B = \{x_1, ...x_K\}$, where $K = Nd$, let $V = \{\mathbf{v}_1, ...\mathbf{v}_K\}$, be the corresponding batch of features obtained as described above. The proposed W-MSE loss is obtained using the MSE computed over all $Nd(d-1)/2$ positive pairs, where constraint 4 is satisfied using the reparameterization of the $\mathbf{v}$ variables with whitened variables $\mathbf{z}$:

$$L_{W-MSE}(V) = \frac{2}{Nd(d-1)} \sum_{(\mathbf{v}_i, \mathbf{v}_j) \in V, pos(i,j)} dist(\mathbf{z}_i, \mathbf{z}_j), \tag{6}$$

where $\mathbf{z} = Whitening(\mathbf{v})$, and:

$$Whitening(\mathbf{v}) = W_V(\mathbf{v} - \boldsymbol{\mu}_V). \tag{7}$$

In equation 7, $\boldsymbol{\mu}_V$ is the mean of the elements in V:

$$\boldsymbol{\mu}_V = \frac{1}{K} \sum_k \mathbf{v}_k, \tag{8}$$

while the matrix W_V is such that: $W_V^\top W_V = \Sigma_V^{-1}$, being Σ_V the covariance matrix of V:

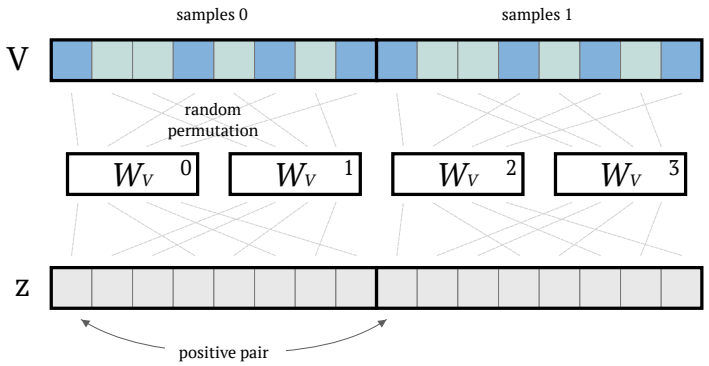

Figure 3: Scheme of batch slicing. V is a batch sliced in $d = 2$ parts, on this scheme v_i and v_{i+8}, $i \in [0, 7]$ represents the same original image ($pos(i, i + 8)$). Each part is equally permuted and additionally sliced, followed by the whitening transform applied independently for each part.

$$\Sigma_V = \frac{1}{K-1} \sum_k (\mathbf{v}_k - \boldsymbol{\mu}_V)(\mathbf{v}_k - \boldsymbol{\mu}_V)^T. \tag{9}$$

For more details on the whitening transform, we refer to Siarohin et al. (2019). This transformation performs the full whitening of each $\mathbf{v}_i \in V$ and the resulting set of vectors $Z = \{\mathbf{z}_1, ..., \mathbf{z}_K\}$ lies in a zero-centered distribution with a covariance matrix equal to the identity matrix (Fig. 1).

The intuition behind the proposed loss is that equation 6 penalizes positives which are far apart from each other, thus leading $g(E(\cdot))$ to shrink the inter-positive distances. On the other hand, since Z must lie in a spherical distribution, the other samples should be "moved" and rearranged in order to satisfy constraint 4 (see Fig. 1).

Batch Slicing. The estimation of the Mean Square Error in equation 6 depends on the whitening matrix W_V, which may have a high variance over consecutive iteration batches $V_t, V_{t+1}, ...$. For this reason, inspired by the resampling methods (Efron, 1982), given a batch V, we slice V in several non-overlapping sub-batches and we compute the whitening matrix independently for each sub-batch. The batch is sliced in d parts, so that each part corresponds to a group of different images. In some cases, even smaller sub-batches are beneficial, the heuristic is the size of the sub-batch should be close to the size of embedding $\times 2$. In this case we choose the permutation randomly (Fig. 3), and it is possible to repeat this operation several times to get a more robust estimate of equation 6.

3.1 Discussion

In a common *instance-discrimination* task (Sec. 2), e.g., solved using equation 2, the similarity of a positive pair ($\mathbf{z}_i^T \mathbf{z}_j$) is contrasted with the similarity computed with respect to all the other samples ($\mathbf{z}_k$) in the batch ($\mathbf{z}_i^T \mathbf{z}_k$, $1 \leq k \leq K, k \neq i$). However, $\mathbf{z}_k$ and $\mathbf{z}_i$, extracted from different image instances, can occasionally share the same semantics (e.g., x_i and x_k are two different image *instances* of the "cat" *class*). Conversely, the proposed W-MSE loss does not force all the instance samples to lie far from each other, but it only imposes a soft constraint (equation 4), which avoids degenerate distributions.

Note that previous work (He et al., 2019; Hénaff et al., 2019; Chen et al., 2020) highlighted that Batch Normalization (BN) may be harmful for learning semantically meaningful representations because the network can "cheat" and exploit the batch statistics in order to find a trivial solution to equation 2. However, our whitening transform (equation 7) is applied only to the very last layer of the network $f(\cdot)$ (see Fig. 2) and it is not used in the intermediate layers, which is instead the case of BN. Hence, our $f(\cdot)$ cannot learn to exploit subtle inter-sample dependencies introduced by batch-statistics because of the lack of other learnable layers on top of the $\mathbf{z}$ features.

Chen et al. (2020) include an additional BN after the projection head $g(\cdot)$, performing centering and scaling of the output features. However, this operation will not prevent the degenerate solution without the contrastive scheme. Indeed, if we just substitute whitening with BN in W-MSE, the

network can easily find a solution where all the dimensions of the embedding represent the same feature. We confirm this behaviour with an additional experiment in which the network converges to zero loss after a few epochs with a low resulting performance.

4 EXPERIMENTS

We test our loss and its competitors on the following **datasets**.

- CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), two small-scale datasets composed of 32×32 images with 10 and 100 classes, respectively.
- Tiny ImageNet (Le & Yang, 2015), a reduced version of ImageNet, composed of 200 classes with images scaled down to 64×64. The total number of images is: 100K (training) and 10K (testing).
- STL-10 (Coates et al., 2011), also derived from ImageNet, with 96×96 resolution images. While CIFAR-10, CIFAR-100 and Tiny ImageNet are fully-labeled, STL-10 is composed of 5K labeled training samples (500 per class) and 100K unlabeled training examples from a similar but broader distribution of images. There are additional 8K labeled testing images.

Setting. The goal of experiments is to compare W-MSE with current state-of-the-art losses, isolating effects of other settings, such as the architecture choice. For this reason, we use the same encoder $E(\cdot)$ ResNet-18 for all experiments. We search for the best hyperparameters for every method (for CIFAR-10 and STL-10, transferring to CIFAR-100 and Tiny ImageNet respectively); we use one configuration, which is optimal for every loss considered. Each reported method uses feature normalization unless otherwise stated. *Contrastive* refers to our implementation of the contrastive loss (equation 2) following details from Chen et al. (2020), with temperature $\tau = 0.5$. *BYOL* is our reproduction of Grill et al. (2020), introduced concurrently with our work. For this method we use the exponential moving average with cosine increasing, starting from 0.99. *W-MSE 2* and *4* are versions with $d = 2$ and $d = 4$ samples per image. For CIFAR-10 and CIFAR-100 sub-batch size is 128, for Tiny ImageNet and STL-10 it is 256. For experiments W-MSE 2 for Tiny ImageNet and STL-10 we use 4 iterations of batch slicing, for all other experiments we use 1 iteration.

For all experiments we use the Adam optimizer (Kingma & Ba, 2014). For CIFAR-10 and CIFAR-100 the training consists of 1000 epochs with learning rate 3×10^{-3}, for Tiny ImageNet – 1000 epochs with learning rate 2×10^{-3}, for STL-10 – 2000 epochs with learning rate 2×10^{-3}. We use learning rate warm-up for the first 500 iterations of the optimizer and 0.2 learning rate drop 50 and 25 epochs before the end. We use a mini-batch size of $K = 1024$ samples. The dimension of the hidden layer of the projection head $g(\cdot)$ is 1024. The weight decay is 10^{-6}. Finally, we use an embedding size of 64 for CIFAR-10 and CIFAR-100, and an embedding of size of 128 for STL-10 and Tiny ImageNet.

As a common practice when using ResNet-like architectures for small-size image resolutions, in all the experiments we have a first convolutional layer with kernel size 3, stride 1 and padding 1. Additionally, in case of CIFAR-10 and CIFAR-100, we remove the first max pooling layer.

Image Transformation Details. We extract crops with random size from 0.2 to 1.0 of the original area and a random aspect ratio from $3/4$ to $4/3$ of the original aspect ratio, which is a commonly used data-augmentation technique. We also apply horizontal mirroring with probability 0.5. Finally, we apply color jitterering with probability 0.8 and grayscaling with probability 0.1.

Evaluation Protocol. The most common evaluation protocol for unsupervised feature learning is based on *freezing* the network encoder ($E(\cdot)$, in our case) after unsupervised pre-training, and then train a supervised *linear classifier* on top of it. Specifically, the linear classifier is a fully-connected layer followed by softmax, which is placed on top of $E(\cdot)$ after removing the projection head $g(\cdot)$. In all the experiments we train the linear classifier for 500 epochs using the Adam optimizer and the labeled training set of each specific dataset, without data augmentation. The learning rate is exponentially decayed from 10^{-2} to 10^{-6}. The weight decay is 5×10^{-6}. We include the accuracy of the k-nearest neighbors classifier (k-nn, $k = 5$). The advantage of this score is that it does not require additional parameters and training, it is deterministic.

Table 1: Classification accuracy (top 1) of a linear and 5-nearest neighbors classifiers of different loss functions and datasets with ResNet-18 encoder.

Method	CIFAR-10		CIFAR-100		STL-10		Tiny ImageNet	
	linear	5-nn	linear	5-nn	linear	5-nn	linear	5-nn
Contrastive	91.80	88.42	66.83	56.56	90.51	85.68	48.84	32.86
BYOL	91.73	89.45	66.60	56.82	91.99	88.64	51.00	36.24
W-MSE 2	91.55	89.69	66.10	56.69	90.36	87.10	48.20	34.16
W-MSE 4	91.99	89.87	67.64	56.45	91.75	88.59	49.22	35.44

Table 2: Classification accuracy (top 1) using Euclidean distance (unnormalized embeddings for contrastive loss) on STL-10.

Method	linear	5-nn
Contrastive	78.00	71.07
BYOL	80.83	74.94
W-MSE 2	89.91	85.56
W-MSE 4	90.40	87.09

4.1 Comparison with the State of the art

Results of experiments are listed in Tab. 1. For W-MSE, 4 samples are generally better than 2. Contrastive loss performs worse in most cases. W-MSE 4 performs best on CIFAR-10 and CIFAR-100, BYOL leads on STL-10 and Tiny ImageNet, although the gap between losses is minor. We include training dynamics (Appendix A) for STL-10 dataset. The plot shows that W-MSE 4 and BYOL have a similar performance during most of the training. However, first 120 epochs BYOL significantly underperforms (e.g., accuracy after 20 epochs: W-MSE 4 79.98, BYOL 73.24), indicating that BYOL requires some "warmup" period; on the other hand, W-MSE performs well from the start. This property is useful in domains requiring rapid adaptation of the encoder, e.g., due to the change of distribution of the input in continual learning or reinforcement learning.

4.2 Euclidean distance

Cosine similarity is a crucial component for both contrastive loss and BYOL, arranging the representation on a hypersphere (Wang & Isola, 2020). However, for W-MSE, whitening transform projects representation onto a spherical distribution, which may provide useful properties without further normalization [2]. We include experiments (Tab. 2) for all considered losses on STL-10 dataset with Euclidean distance, replacing equation 5 with $dist(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2^2$, with unnormalized embeddings and $\tau = 1$ for contrastive loss. These experiments show a strong decline for the contrastive loss and BYOL, while the performance of W-MSE decreases slightly, indicating that in some cases W-MSE can be used with Euclidean distance.

5 Conclusion

In this paper, we have proposed a new self-supervised loss, W-MSE, which is alternative to common loss functions used in the field. Differently from the triplet loss and the contrastive loss, both of which are based on comparing an instance-level similarity against other samples, W-MSE computes only the intra-positive distances, while using a whitening transform to avoid degenerate solutions. Naturally, such loss encourages to use more positive pairs in the batch. We empirically show that W-MSE achieves results constantly better than the popular contrastive loss. Moreover, W-MSE is competitive with concurrently introduced BYOL, without requiring special additional networks.

[2] Note that the arrangement on the hypersphere and the spherical distribution are different concepts: the first concerns each element independently, while the second applies to the whole set.

REFERENCES

Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

Dariusz Dereniowski and Kubale Marek. Cholesky factorization of matrices in parallel and ranking of graphs. In *5th Int. Conference on Parallel Processing and Applied Mathematics*, 2004.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL,*, 2019.

Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *NeurIPS*, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

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, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

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

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

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

Olivier J. Hénaff, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv:1905.09272*, 2019.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *CVPR*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

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

Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

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

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016.

Mirco Ravanelli and Yoshua Bengio. Learning speaker representations with mutual information. *arXiv:1812.00271*, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and coloring transform for GANs. In *International Conference on Learning Representations*, 2019.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv:1907.13625*, 2019.

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 2020.

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

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv:1805.01978*, 2018.

A APPENDIX

Fig. 4 and 5 show training dynamics for each considered loss. Charts are smoothed with a 0.3 moving average for readability (curves before smoothing are shown semi-transparent).

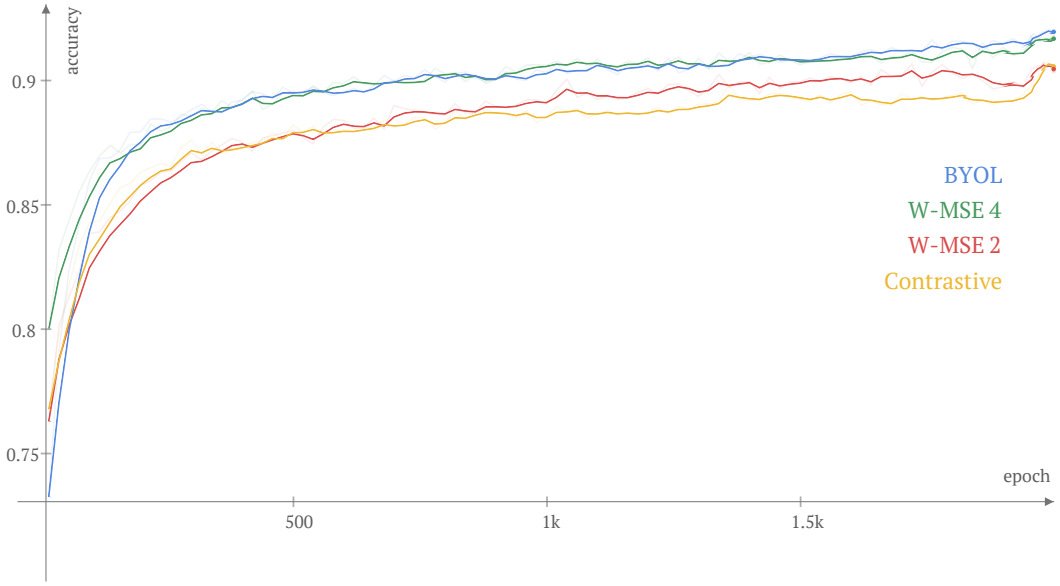

Figure 4: Training dynamics on STL-10 dataset for linear classifier

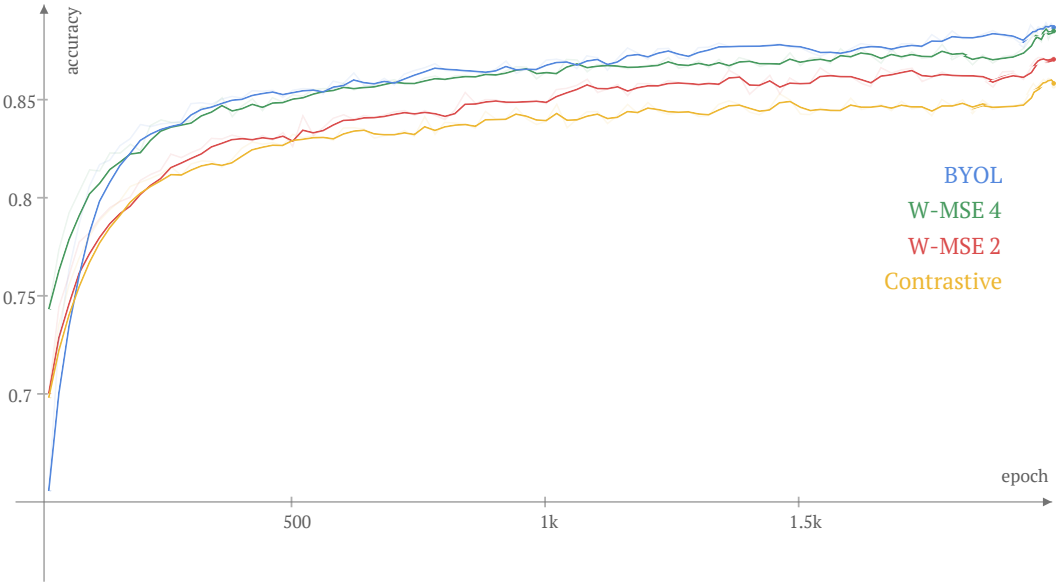

Figure 5: Training dynamics on STL-10 dataset for 5-nn classifier