On Feature Decorrelation in Self-Supervised Learning

Tianyu Hua*[1,2], Wenxiao Wang*[1], Zihui Xue[2,3], Sucheng Ren[2,5], Yue Wang[4], and Hang Zhao†[1,2]

[1]Tsinghua University [2]Shanghai Qi Zhi Institute
[3]UT Austin [4]MIT [5]South China University of Technology

Abstract

In self-supervised representation learning, a common idea behind most of the state-of-the-art approaches is to enforce the robustness of the representations to predefined augmentations. A potential issue of this idea is the existence of completely collapsed solutions (i.e., constant features), which are typically avoided implicitly by carefully chosen implementation details. In this work, we study a relatively concise framework containing the most common components from recent approaches. We verify the existence of ***complete collapse*** *and discover another reachable collapse pattern that is usually overlooked, namely* ***dimensional collapse****. We connect dimensional collapse with strong correlations between axes and consider such connection as a strong motivation for* ***feature decorrelation*** *(i.e., standardizing the covariance matrix). The gains from feature decorrelation are verified empirically to highlight the importance and the potential of this insight.*

1. Introduction

Deep learning is prevailing in a wide range of domains, including computer vision [20], natural language processing [13] and speech recognition [47], while the utility of the most classical, supervised methods are sometimes restricted by limited or costly data labeling. Recently, self-supervised learning has proven capable of offering visual representations with high utility and therefore reducing the need for massive annotations. The past year has witnessed significant advancements in this field: A line of work focuses on determining augmentations that better suit the self-supervised fashion, including revisiting typical augmentations [43], using adversarial perturbations [24, 32, 23], and searching augmentation policies [36]; A line of work alters the sampling strategy to expand the source of positive pairs [49, 2, 44] and calibrate the contribution of negative samples [44, 38, 27]; Another line of work uses clustering-based mechanisms to characterize the relation of cross-sample views [5, 6, 1, 7]. Despite the technical variety, there is one high-level idea that remains in most if not all of the recent approaches, learning representations that are robust to augmentations [35, 8, 19, 43].

This idea is fairly intuitive but it does not rule out trivial, collapsed solutions by design. In consequence, existing work must incorporate a way that helps to mitigate the issue

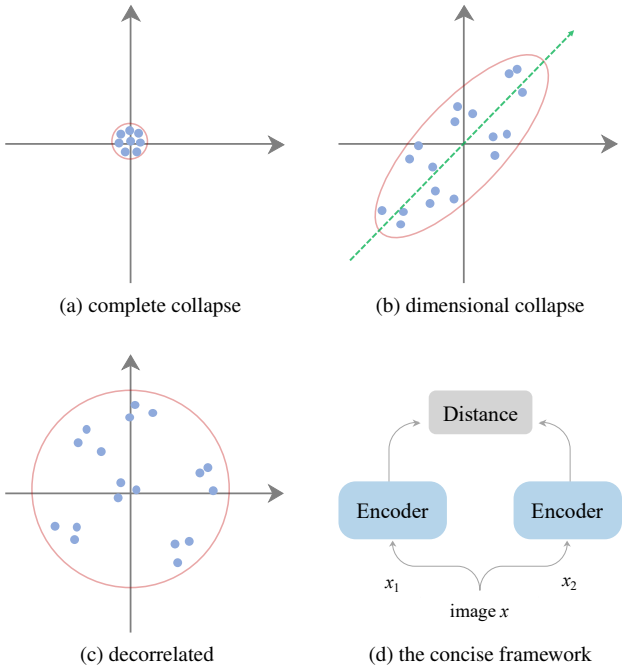


Figure 1: An overview of the key components of this work: 1a and 1b are two reachable collapse patterns in self-supervised settings; 1c is an illustration of the goal of feature decorrelation; 1d is a sketch of the concise framework used in this work.

*Equal contribution.
†Corresponding to hangzhao@mail.tsinghua.edu.cn.

of feature collapsing. Complete collapse, where the representations collapse into a constant as in Figure 1a, is the most well-known type of collapse and is addressed differently in existing work with carefully chosen implementation details: To name a few, SimCLR [8] and Uniformity[45] use losses that maximize the distance between different samples; SwAV [7] includes an additional online clustering branch that clusters data into a predefined number of groups; BYOL [18] relies on a predictor structure, a properly inserted stop-gradient operator and a momentum encoder; SimSiam [10] simplifies the framework of BYOL by removing the momentum encoder. Given their success in preventing complete collapse, the study of other potential collapse issues in self-supervised learning has been ignored.

Meanwhile, feature decorrelation appears to be a valuable idea in the field of machine learning: In discriminative tasks, [12, 48] introduce in objective functions additional terms regularizing correlation matrices and [25, 26] develop normalization layers standardizing covariance matrices to obtain higher accuracy; In generative tasks, it is through feature decorrelation that [41] produces more realistic synthesized images and [39] offers better utility of domain adaptation.

In this work, we revisit the collapse issue of self-supervised learning and show how the idea of feature decorrelation helps to resolve the issue and improve the utility, using a framework presented in Figure 1d that contains the most common components of existing approaches. Our contribution includes:

- We verify the existence of complete collapse in self-supervised settings and address it successfully by standardizing variance.
- We discover another reachable collapse pattern ignored by existing works, namely dimensional collapse.
- We reveal the connection between dimensional collapse and strong correlations, which leads to the idea of standardizing covariance (*i.e.* feature decorrelation).
- Empirically, the performance gains from feature decorrelation in a wide range of settings confirm the importance and the potential of this insight.

2. Related Work

Contrastive learning. Contrastive approaches learn representations by maximizing agreement between two augmented views of a sample (*i.e.*, positive pairs) and disagreement of views from different samples (*i.e.* negative pairs). Following this idea, many methods have been developed [35, 22, 46, 21, 19, 8, 45]. As they benefit from a large number of negative samples, contrastive learning methods require a memory bank [46], a queue [19] to store negative samples, or large batch sizes [8] to work well. This leads to the question of whether using negative samples is necessary.

Clustering. Clustering-based methods partially answer this question. They discriminate between groups of images with similar features instead of individual images [5, 6, 1, 7]. SwAV [7] clusters data and enforces consistency between cluster assignments produced from different views of the same sample. However, these methods require a costly clustering phase and large batches to have a sufficient number of samples for clustering [18, 10].

BYOL and SimSiam. Another recent line of work achieves remarkable results by only using positive samples. BYOL [37] proposes an online network along with a target network, where the target network is updated with a moving average of the online network to avoid collapse. Contrary to them, SimSiam [10] demonstrates that a predictor network and a properly inserted stop-gradient operator are the crucial components in preventing collapse. Tian *et al*. provide an analysis of how various factors involved in BYOL and SimSiam work together to prevent collapse [42].

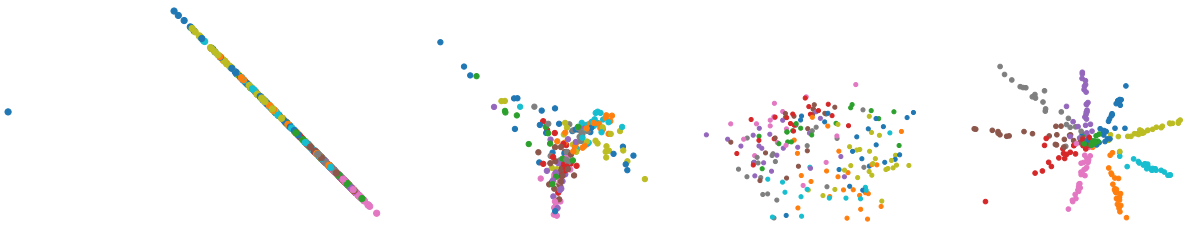
Normalization. Different from previous works that attribute collapse prevention to "asymmetry", *i.e.*, a predictor network and stop gradient, we propose a new angle to understand collapse in this work. Based on this view, we introduce normalization techniques in supervised learning to the task of learning representations without negative pairs. Batch Normalization (BN) [28] is the first to perform normalization per mini-batch in a way that supports back-propagation, and has shown remarkable performance in training deep neural networks. The idea of BN is to center and scale activations. Another normalization technique, Decorrelated Batch Normalization (DBN) [25] proposes to whiten activations within each mini-batch. In this work, we demonstrate that in the context of self-supervised learning, BN encounters dimensional collapse while DBN effectively avoids all kinds of collapse.

Two concurrent works [14, 51] explore similar ideas to ours for preventing collapses. W-MSE [14] whitens feature representations within each batch via Cholesky decomposition. Barlo Twins [51] enforces the cross-correlation matrix between outputs of a positive pair to be close to identity, using an additional loss function. These attempts corroborate the potential of feature decorrelation and highlight the necessity of our findings towards understanding and addressing feature collapses.

3. Main Results

In this section, we will take a close look at two collapse patterns in self-supervised learning settings. We will show how the first one, a well-known collapse pattern termed **complete collapse**, is addressable with BN [28] since it is associated with vanishing variances.

Furthermore, with complete collapse avoided, we discover another reachable collapse pattern overlooked by existing work, namely **dimensional collapse**. We relate di-



(a) baseline: complete collapse (b) Batch Normalization: dimensional collapse (c) Decorrelated BN: decorrelated space (d) SimCLR (e) Supervised

Figure 2: Direct visualization of 2-dimensional projection spaces on CIFAR-10. Different colors correspond to different classes. Figure 2a, 2b and 2c are from the concise framework we use. For completeness, we visualize the 2-dimensional projection spaces of SimCLR (by setting the output dimension of the projector to be 2) and a supervised baseline (by letting the penultimate layer to contain 2 neurons) in Figure 2d and 2e.

mensional collapse to strong correlations between axes and show with DBN [25] that standardizing covariance matrix helps in alleviating dimensional collapse.

We also introduce in this section an easy add-on to DBN that enforces further decorrelation, which will be compared empirically with DBN in Section 4 to support the importance and the potential of feature decorrelation. We refer to DBN with this add-on as Shuffled-DBN.

In the last part of this section, we include elaborations of some details that are postponed for coherence.

3.1. Preliminary

We utilize here a relatively concise framework for self-supervised representation learning as follows, which contains only the most common components of modern self-supervised approaches:

Definition 1 (Concise Framework) *In the concise framework, given the training data distribution $\mathcal{D}$ and the augmentation distribution $\mathcal{T}$, the model parameter θ is trained to maximize/minimize the objective function with the following form:*

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ T_1,T_2\sim\mathcal{T}}} \ell(f_\theta(x_1),\ f_\theta(x_2)),$$

where f_θ is the encoder that contains a backbone and a projector, $x_1 = T_1(x)$, $x_2 = T_2(x)$, and ℓ is the similarity/distance function. A sketch of this framework is presented in Figure 1d.

Unless otherwise specified, squared error $\ell(z_1, z_2) = \|z_1 - z_2\|_2^2$ is used as the distance function. We will elaborate the choice of ℓ in Section 3.4.1.

3.2. Reachable Collapse Patterns and Their Indicators

To build up intuitions, we now apply to CIFAR-10 a specific realization of the concise framework, which we refer to as the baseline: The encoder f_θ is a ResNet-18 backbone plus a projector MLP with an output dimension of 2 and two hidden layers with 64 neurons in each. ReLU activation and BN are appended to both hidden layers of the projector.

Here we set the output dimension of the projector to be 2 for easy visualization. We will show later that our findings remain in high-dimensional projection space.

The resulted representation yields an accuracy of only 28.56% in linear evaluation and we visualize in Figure 2a the projection space (*i.e.* $f_\theta(T(x))$ where $x \sim \mathcal{D}, T \sim \mathcal{T}$).

In Figure 2a, we observe a projection space that collapses to a single point, which we refer to as complete collapse. When it happens, almost no gradient can be propagated back through the projection space (since $\nabla_{f_\theta(T(x))}\ell \approx 0$) to influence the learned representation and therefore its utility is compromised.

Complete collapse is a widely known type of collapse in representation learning, and it is associated with **vanishing variances**. Accordingly, using BN in the projection space to standardize variance can be a way to mitigate complete collapse.

Definition 2 (Batch Normalization [28]) *For a Batch Normalization (BN) layer that takes as its input a batch of D-dimensional vectors $X = (x_1, \cdots, x_B) \in \mathcal{R}^{D\times B}$, its output is a batch of vectors $Y = (y_1 \cdots, y_B) \in \mathcal{R}^{D\times B}$, computed as follows:*

$$y_{b,d} = \frac{x_{b,d} - \mu_d}{\sqrt{\sigma_d^2 + \epsilon}} \cdot \gamma_d + \beta_d$$

for all $b \in \{1, \cdots, B\}$ and $d \in \{1, \cdots, D\}$, where γ, β are learnable affine parameters, ϵ is a small constant originally proposed for numerical stability. In training time, μ_d, σ_d^2 are mean and variance computed over the d-th row of the input batch X, and in inference time, running estimations from training time are used.

We append to the projector of the baseline an additional BN layer with no affine parameter and $\epsilon = 0$ (*i.e.* $y_{i,j} = \frac{x_{i,j}-\mu_j}{\sqrt{\sigma_j^2}}$, whose necessity will be elaborated in Section 3.4.2) and visualize the projection space in Figure 2b. The corresponding representation yields an accuracy of 69.52% in linear evaluation, significantly improved over 28.56% obtained by the completely collapsed baseline.

With complete collapse resolved, we notice another usually overlooked collapse pattern in the projection space, termed dimensional collapse, for which the projected features collapse into a low-dimensional manifold such as the single line in Figure 2b. Dimensional collapse can harm utility and should be addressed appropriately. By definition, dimensional collapse is associated with **strong correlations** between axes. As a sanity check, we adapt DBN [25] to standardize the covariance matrix for mitigation of this issue.

Definition 3 (Decorrelated Batch Normalization [25]) *The Decorrelated Batch Normalization (DBN) layer with a group size G takes as its input a batch of D-dimensional vectors $X = (x_1, \cdots, x_B) \in \mathcal{R}^{D\times B}$ and its output is a batch of vectors $Y = (y_1 \cdots, y_B) \in \mathcal{R}^{D\times B}$ computed as follows:*

$$Y^{[h]} = ZCA(X^{[h]}),$$

where $X^{[h]} = \left(\left(X_{(h-1)\cdot G+1}\right)^T, \cdots, \left(X_{h\cdot G}\right)^T\right)^T \in \mathcal{R}^{G\times B}$ *and* $Y^{[h]} = \left(\left(Y_{(h-1)\cdot G+1}\right)^T, \cdots, \left(Y_{h\cdot G}\right)^T\right)^T \in \mathcal{R}^{G\times B}$. *In other words, DBN divides the D feature dimensions into groups of size G and applies ZCA whitening to each group independently.*

Definition 4 (ZCA Whitening [3]) *ZCA Whitening takes as its input a batch of D-dimensional vectors $X = (x_1, \cdots, x_B) \in \mathcal{R}^{D\times B}$ and its output is a batch of vectors $Y = (y_1 \cdots, y_B) \in \mathcal{R}^{D\times B}$ computed as follows:*

$$Y = Q\Lambda^{-\frac{1}{2}}Q^T\hat{X},$$

where $\hat{X}$ is X with rows normalized to zero-mean (i.e. $\hat{X}_{d,b} = X_{d,b} - \frac{1}{B}\sum_{k=1}^{B} X_{d,k} = x_{b,d} - \frac{1}{B}\sum_{k=1}^{B} x_{k,d}$), $\Lambda \in \mathcal{R}^{D\times D}$ is a diagonal matrix filled with the eigenvalues of $\Sigma = \hat{X}\hat{X}^T$ and $Q \in \mathcal{R}^{D\times D}$ is the corresponding orthonormal eigenvectors (i.e. $\Sigma = Q\Lambda Q^T$). ZCA assumes $\Sigma = \hat{X}\hat{X}^T \in \mathcal{R}^{D\times D}$ is full-rank.

The rows of the ZCA's output Y are zero-mean, and therefore the corresponding covariance matrix is

$$\begin{aligned} YY^T =& Q\Lambda^{-\frac{1}{2}}Q^T\hat{X}\hat{X}^TQ\Lambda^{-\frac{1}{2}}Q^T \\ =& Q\Lambda^{-\frac{1}{2}}Q^TQ\Lambda Q^TQ\Lambda^{-\frac{1}{2}}Q^T \\ =& Q\Lambda^{-\frac{1}{2}}\Lambda\Lambda^{-\frac{1}{2}}Q^T \\ =& QQ^T = I. \end{aligned}$$

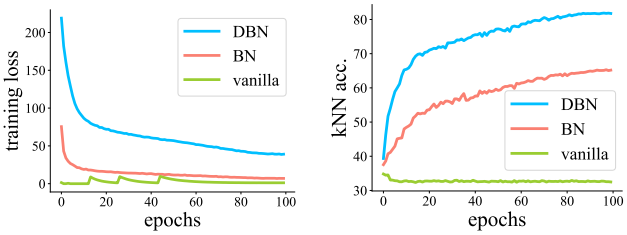


Figure 3: A comparison of learning processes with different variants of the concise framework. For both collapse patterns, collapsed variants optimize loss function easily but offer representations with degraded utility.

	acc. (%)	std.	corr.	loss
vanilla	35.44	**0.00**	0.13	0.00
BN	70.85	1.00	**0.99**	7.01
DBN	84.41	1.00	0.00	39.04

Table 1: A comparison of the eventual representations with different variants of the concise framework: **acc.** denotes the accuracy in linear evaluation; **std.** denotes the average standard deviation over 128 dimensions of the projected features; **corr.** denotes the average correlation strength (*i.e.* the average of the absolute values of non-diagonal entries of the correlation matrix) of the projected features; **loss** denotes the training loss. The group size of DBN is 128.

Thus DBN standardizes the covariance matrices of dimension groups to alleviate dimensional collapse issues.

We visualize in Figure 2c the projection space with a DBN layer (group size $G = 2$) appended to the projector of the baseline. The corresponding representation offers an accuracy of 72.45% in linear evaluation, which reveals already, in this 2-dimensional case, a non-negligible gap from 69.52% offered by the dimensionally collapsed one. It corroborates the importance of feature decorrelation to self-supervised representation learning.

The collapse patterns we observed remain reachable with a high-dimensional projection space and remain linked respectively with vanishing variances and strong correlations. In Figure 3 and Table 1, we include a comparison of these variants when the projector is a 2-layer MLP with 128 hidden neurons and 128-dimensional outputs.

In this comparison, we observe vanishing variances (through std.) with the vanilla framework and strong correlations (through corr.) with BN, which serve as signs of complete collapse and dimensional collapse, respectively. Another observation is that the utility gain by addressing these collapse patterns enlarges with an increased dimension of projection spaces. This observation further corroborates the potential of feature decorrelation.

3.3. Further Decorrelation, Further Gains

We show in the previous section that feature decorrelation (*i.e.* standardizing covariance matrix) alleviates an overlooked pattern of collapse named dimensional collapse and therefore improves utility.

However, the dimensional collapse issue partially remains since DBN introduces a grouping strategy (whose necessity is elaborated in Section 3.4) that standardizes only covariances within each dimension group. To reveal the potential of feature decorrelation, we propose a variant of DBN with one easy add-on, namely Shuffled-DBN. In this section, we show that Shuffled-DBN offers further decorrelation. A more thorough evaluation of the further gains from the further decorrelation is included in Section 4.

Definition 5 (Shuffled-DBN) *The Shuffled-DBN layer with a group size G takes as its input a batch of D-dimensional vectors $X = (x_1, \cdots, x_B) \in \mathcal{R}^{D\times B}$ and its output is a batch of vectors $Y = (y_1 \cdots, y_B) \in \mathcal{R}^{D\times B}$ computed as follows:*

$$Y = \mathcal{P}^{-1}\left(DBN_G(\mathcal{P}(X))\right),$$

where $\mathcal{P}$ is a random D-order permutation, $\mathcal{P}(X)$ is obtained by rearranging rows of X according to $\mathcal{P}$ and $\mathcal{P}^{-1}(X)$ is obtained by rearranging rows according to the inverse permutation of $\mathcal{P}$.

In other words, Shuffled-DBN permutes the D feature dimensions randomly before applying DBN with the same group size G and reverses the permutation for outputs.

Intuitively, Shuffled-DBN enforces further decorrelation since now each dimension is whitened with another $G-1$ randomly chosen dimensions rather than fixed ones, which standardizes the covariance matrix better. We verify the intuition empirically and include the results in Figure 4, where Shuffled-DBN offers lower correlation strength, less dimensional collapse, and better utility as expected. These support our claim regarding further decorrelation of Shuffled-DBN.

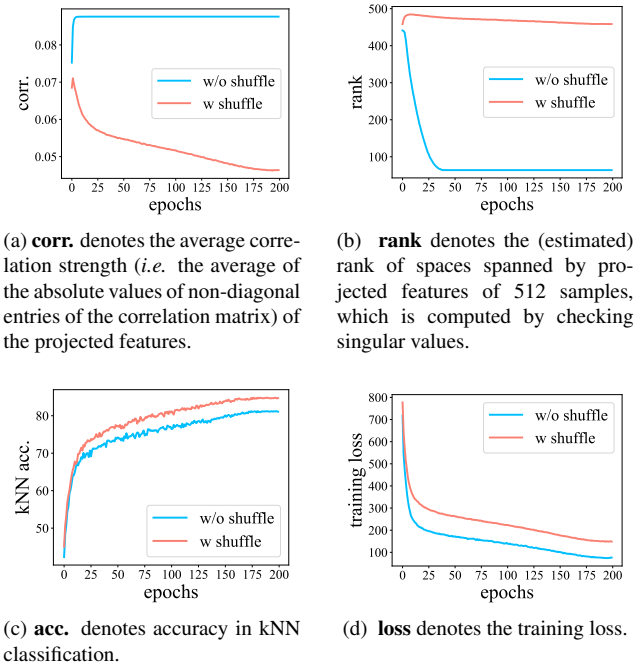
3.4. Details that Matter

In this section, we will clarify details regarding choices and explanations that are previously skipped for coherence, including the choice of the objective ℓ, the detailed setup of BN and the role of grouping in DBN.

3.4.1 The Choice of The Objective ℓ

Here we will explain why we choose squared error $\ell_{SE}(z_1, z_2) = \|z_1 - z_2\|_2^2$ as our default setting instead of cosine similarity $\ell_{cos}(z_1, z_2) = \frac{z_1^T z_2}{\|z_1\|_2\|z_2\|_2}$, which is more popular in self-supervised representation learning.

The main difference between maximizing cosine similarity ℓ_{cos} and minimizing squared error ℓ_{SE} is whether



(a) **corr.** denotes the average correlation strength (*i.e.* the average of the absolute values of non-diagonal entries of the correlation matrix) of the projected features.

(b) **rank** denotes the (estimated) rank of spaces spanned by projected features of 512 samples, which is computed by checking singular values.

(c) **acc.** denotes accuracy in kNN classification.

(d) **loss** denotes the training loss.

Figure 4: A comparison of DBN (*i.e.* w/o shuffle) and Shuffled-DBN (*i.e.* w shuffle). Compared with DBN, Shuffled-DBN standardizes covariance matrix better (lower corr.), mitigates dimensional collapse more thoroughly (higher rank), and offers better utility (higher acc.). The group size is 64 in both cases, and an MLP with two hidden layers (with respectively 512 and 1024 neurons) and an output dimension of 512 is used as the projector.

or not the vectors are normalized to unit L_2-norms, since $\ell_{SE}(z_1, z_2) = 2 - 2\ell_{cos}(z_1, z_2)$ when $\|z_1\|_2 = \|z_2\|_2 = 1$.

Such normalization may conflict with feature decorrelation. With squared error, for $i \in \{1, 2\}, j \in \{1, \cdots, D\}$ where $z_1, z_2 \in \mathcal{R}^D$, we have

$$\frac{\partial \ell_{SE}}{\partial z_{i,j}} = 2(z_{i,j} - z_{3-i,j}),$$

which involves only the j-th dimension itself.

However, with cosine similarity, for $i \in \{1, 2\}, j \in \{1, \cdots, D\}$ where $z_1, z_2 \in \mathcal{R}^D$, we have

$$\frac{\partial \ell_{cos}}{\partial z_{i,j}} = \frac{z_{3-i,j}}{\|z_{3-i}\|_2} \cdot \left(\frac{1}{\|z_i\|_2} - \frac{z_{i,j}^2}{\|z_i\|_2^3}\right) = \frac{z_{3-i,j}\sum_{k\neq j} z_{i,k}^2}{\|z_{3-i}\|_2\|z_i\|_2^3},$$

which may depend heavily on other dimensions because of the normalization. This may introduce unnecessary interference in the projection space.

We suggest removing such normalization is favorable in feature decorrelation. Experimentally, Shuffled-DBN fails in decorrelation with cosine similarity as the objective.

ϵ	learnable affine	acc.(%)
0	No	70.85
0.1	No	34.47
0	Yes	10.00

Table 2: A comparison of BN with different setups. Both the learnable affine transform and non-negligible ϵ are detrimental to eventual utility since they compromise variance standardization.

3.4.2 The Detailed Setup of BN

Here we will explain the detailed setup in using BN to avoid complete collapse. We include an empirical comparison of BN with different setups in Table 2, where both the learnable affine transform and a non-negligible ϵ are detrimental to the utility of the learned representation.

The learnable affine transformation nullifies variance standardization simply because the variance of the output of BN scales linearly with the scaling parameter γ.

As for the ϵ designed originally for numerical stability, one should notice that for a given batch of inputs $X \in \mathcal{R}^{D\times B}$ with variance $\sigma^2 \in \mathcal{R}^D$, the variance of the d-th dimension of its output is in fact

$$\hat{\sigma}_d^2 = \frac{\sigma_d^2}{\sigma_d^2 + \epsilon} = 1 - \frac{\epsilon}{\sigma_d^2 + \epsilon},$$

which is strictly monotonically increasing with σ_d as long as $\epsilon > 0$ and therefore vanishing variance remains as a trivial, reachable solution.

Note that in Table 2, we report in the comparison a setting with $\epsilon = 0.1$, which is greater than typical choices and is used only as a proof of concept.

3.4.3 The Role of Grouping in DBN

The grouping strategy in DBN has two major benefits, one is for flexibility, and another is for efficiency.

Recall that ZCA Whitening works only with the assumption that $\Sigma = \hat{X}\hat{X}^T \in \mathcal{R}^{D\times D}$ is full-rank. Otherwise, no linear transform on the features can result in a fully standardized covariance matrix, *i.e.* an identity matrix I, since I is full-rank.

To have Σ a matrix with rank D, $\hat{X} \in \mathcal{R}^{D\times B}$ must have a rank of at least D. Besides, since each row of $\hat{X}$ is zero-mean, we have its rank to be bounded by $B - 1$, which indicates that the minimum batch size B allowed is at least $D + 1$. This greatly limits the flexibility of ZCA Whitening as one has to either restrict the dimension of the feature space or scale the batch size linearly with it. With grouping, the batch size only has to scale with the group size G.

Another relatively minor benefit is the improved efficiency. Without grouping, a single pass of ZCA Whitening has a computational cost of $O(BD^2)$, with B the batch size and D the number of dimensions. In comparison, DBN with a group size of G requires only a cost of $O(BDG)$.

4. Evalution

4.1. Experimental Setup

• **Benchmarks.** We conduct extensive experiments on 5 popular benchmarks. **CIFAR-10** and **CIFAR-100** [30] are two small-scale image datasets composed of 32×32 small images with 10 and 100 classes, respectively. **STL-10** [11] and **Tiny ImageNet** [31] are both medium-size datasets derived from the ImageNet dataset [40]. The STL-10 dataset is composed of 96 × 96 resolution images of 10 classes. For each class, STL-10 has 500 labeled training samples (5K labeled training samples in total) and 800 labeled samples for testing. An additional 100K unlabeled training images are sampled from a wider range of images than labeled ones. The Tiny ImageNet dataset has 200 classes and comprises 100K training data and 10K testing data, with 64 × 64 resolutions. **ImageNet ILSVRC-2012** is a popular large-scale image dataset of 1000 classes and 1.28M training images. It has 50K images for validation and 150K for testing.

• **Optimizer and learning rate**. Large-batch optimizers such as LARS [50] are commonly used in self-supervised contrastive pre-training for visual representation learning [8, 18, 7]. However, recent studies [15, 16] indicate such adaptive gradient optimizers might regularize the network the same way batch norm does. To separate the inherent normalization properties of optimizers, we use SGD for pre-training. We set our base learning rate to be 0.02 for experiments on small and medium-sized datasets and 0.06 for large-scale datasets. We linearly scale the learning rate according to the batch size: $\frac{\text{base lr}\times\text{batch size}}{256}$ [17]. The learning rate is scheduled to a cosine decay rate and 5 warm-up epochs [34]. We keep the momentum parameter to be 0.9. The weight decay rate is 0.001 for small and medium-sized datasets and 1×10^{-4} for ImageNet.

• **Encoder-backbone.** ResNet-18 is adopted as the backbone of our encoder on small and medium datasets. For CIFAR-10 and CIFAR-100, we use the CIFAR variant of ResNet-18 [20, 10], the first max-pooling layer of which is removed, and the kernel size of the first convolution layer is 3. For medium-size datasets STL-10 and Tiny ImageNet, only the max-pooling layer is disabled following [14, 10]. We adopt a ResNet-50 as the encoder for large-scale ImageNet experiments. We remove the last fully connected layer in ResNet-18 and ResNet-50 models and treat the features after global average pooling as inputs to the projector.

• **Encoder-projector.** The projector is a 3-layer projection MLP. BN and ReLU activations are applied to all hidden

	CIFAR-10	CIFAR-100	STL-10	Tiny ImageNet
SimCLR [8]	86.96	55.86	85.50	42.65
BYOL [37]	86.65	59.33	85.59	42.75
SimSiam [10]	86.31	59.44	**86.55**	41.58
Barlow Twins [51]	89.02	62.84	85.43	45.33
DBN	86.32	56.49	82.36	40.37
Shuffled-DBN	**89.50**	**62.95**	86.02	**45.96**

Table 3: Top-1 accuracies(%) of DBN and Shuffled-DBN in linear evaluation with 200-epoch pretraining. For completeness and reference, we include results of some representative methods from our reproduction. For a fair comparison, we use the same projector and augmentations as we describe in Section 4.1 for all methods in the reproduction.

	CIFAR-10	CIFAR-100
SimCLR	75.05	50.42
BYOL	78.63	51.44
SimSiam	78.77	50.71
Barlow Twins	79.54	**57.22**
DBN	78.60	52.95
Shuffled-DBN	**80.62**	57.17

Table 4: Top-1 accuracies(%) of DBN and Shuffled-DBN in linear evaluation on CIFAR-10 and CIFAR-100 with 200-epoch pretraining on Tiny ImageNet. For completeness and reference, we include results of some representative methods from our reproduction. For a fair comparison, we use the same projector and augmentations as we describe in Section 4.1 for all methods in the reproduction.

layers of the projector. We set the hidden dimensions twice of the input dimension and keep the output dimension identical to the input dimension. Finally, we normalize the output using our Shuffled-DBN layer. Unless otherwise specified, we set the group size of Shuffled-DBN to be one-half of the batch size.

• **Data augmentation.** We adopt several common data augmentations and compose them stochastically: (a) random scaling and cropping with a scaling factor chosen between $[0.2, 1.0]$; (b) random horizontal flipping with a probability of 0.5; (c) color distortion with a probability of 0.8; (d) color dropping (*i.e.*, randomly convert images to grayscale with 20% probability for each image); (e) random gaussian blur for medium and large-size datasets.

• **Training and Evaluation** We evaluate the quality of the pre-trained representations by training a supervised linear classifier on the frozen representations, following a common protocol. We perform unsupervised pre-training on the train set for 200 epochs. Then we freeze the features and train a supervised linear classifier, *i.e.*, a fully-connected layer followed by a softmax layer, on the extracted features. Specifically, we train the linear layer on top of the global average pooling features of a ResNet for 100 epochs. To test the classifier, we use the center crop of the test set and computes accuracy according to predicted outputs. We train the classifier with a base learning rate of 30, no weight decay, a momentum of 0.9, and a batch size of 256. Note that we only train the classifier over the labeled split of STL-10 since the majority of STL-10 training data is unlabeled. We report the validation accuracy for ImageNet.

dim.	64	128	256	512	1024
DBN	77.17	82.15	82.42	82.91	84.39
Shuffled-DBN	82.92	83.19	84.54	86.02	87.22

Table 5: Top-1 accuracies(%) in linear evaluation of DBN and Shuffled-DBN on CIFAR-10 with different numbers of output dimension for a 3-layer MLP projector (**dim.**): In all cases, we use a group size of 32 and a batch size of 256; The hidden layers of the projector contain 1024 neurons each.

4.2. Gains from Further Decorrelation

In this section, to verify the gains from further decorrelation empirically, we have both DBN and Shuffled-DBN evaluated on multiple benchmarks and have the results reported in Table 3. Through further decorrelation, Shuffled-DBN outperforms DBN on all 4 benchmarks with performances competitive to the best of all evaluated methods, which supports the claim strongly.

In Table 4, we also include a comparison of the generalizability of DBN and Shuffled-DBN, by evaluating representations pretrained with Tiny ImageNet on CIFAR-10 and CIFAR-100. With further decorrelation, Shuffled-DBN generalizes better than DBN in both cases, achieving performances competitive to the best one among all evaluated methods, just as in the prior setting.

In addition, we study the gains from further decorrelation varying the number of dimensions for the projection space. The results are in Table 5, from which one sees that further decorrelation yields further gains consistently, regardless of

batch size	32	64	128	256	512
Shuffled-DBN	88.25	**89.17**	**89.31**	**88.82**	87.92
Barlow Twins	86.89	87.98	88.21	87.57	85.19
BYOL	**88.37**	88.44	87.64	85.72	82.63
SimCLR	85.42	87.41	87.40	87.70	**87.98**
SimSiam	86.84	87.88	86.47	79.02	67.74

Table 6: The top-1 accuracy(%) of Shuffled-DBN and our own reproduction of Barlow Twins, BYOL, SimCLR and SimSiam at 200 epochs under linear evaluation on CIFAR-10. The training and evaluation configurations are the same. 2-layer MLP projectors with hidden dimension and output dimension to be 1024 and 512 are used for all experiments.

group size	16	32	64	128
kNN acc.	83.41	85.93	87.05	87.59
linear acc.	85.52	87.69	88.75	88.29

Table 7: Accuracies(%) of Shuffled-DBN on CIFAR-10 with different group size. **kNN acc.** denotes accuracy in kNN classification. **linear acc.** denotes accuracy in linear evaluation. The output dimension of the projector is 512.

the specific choice of the projector's output dimensions.

Are the aforementioned gains generic varying the batch size? To answer this, we conduct an ablation study regarding the gains of feature decorrelation while using different batch sizes. The results are in Table 6, where Shuffled-DBN offers top utilities in all cases, supporting strongly the generality of the gains from decorrelation.

4.3. Varying Decorrelation Strength

Another way to verify the gains from further decorrelation is to vary the decorrelation strength of Shuffled-DBN, which we achieve here by varying the group size G: The larger the group size G is, the stronger the decorrelation strength will be.

We report in Table 7 the results of such ablation study. We observe that the overall trend is consistent with our expectation in both kNN classification and linear evaluation: The utility improves with a stronger decorrelation strength.

4.4. Feature Decorrelation on ImageNet

The accuracy in linear evaluation on ImageNet has become a de facto metric of visual features learned in self-supervised fashions. While the differences in both accessible computational resources and implementation details (*e.g.* resources for hyperparameter tuning) are detrimental to the fairness of a direct comparison, we present in Table 8 the evaluation on ImageNet and consider it as a nice addition to compare feature decorrelation with representative methods. For completeness, we also include in Table 9 the top-1 and top-5 accuracies of Shuffled-DBN for checkpoints in the middle.

method	batch size	top-1
InstDisc [46]	256	58.5
LocalAgg [52]	128	58.8
MoCo [19]	256	60.6
SimCLR [8]	256	61.9
CPC v2 [35]	512	63.8
PCL v2 [33]	256	67.6
MoCo v2 [9]	256	67.5
MoCHi [29]	512	68.0
PIC [4]	512	67.6
AdCo [24]	256	68.6
Shuffled-DBN	512	65.18

Table 8: Top-1 accuracies(%) in linear evaluation on ImageNet with the ResNet-50 backbone and 200 epochs of pretraining. The table are mostly inherited from [24]. Feature decorrelation in the concise framework acheives sub-optimal performance.

epoch	10	20	50	100	150	200
Top-1	49.12	54.32	57.43	59.04	62.24	65.18
Top-5	72.34	76.93	79.24	80.62	82.88	85.32

Table 9: Top-1 and top-5 accuracies of Shuffled-DBN in linear evaluation on ImageNet varying pre-training epochs.

On ImageNet, although Shuffled-DBN does not achieve state-of-the-art performances, it remains promising given that it achieves sub-optimal utility in a concise framework (*i.e.*, with no predictor, no momentum encoder and no other special implementation detail).

5. Conclusion

In this work, we study the feature collapsing issues in self-supervised learning. Firstly, we verify the existence of complete collapse and address it by standardizing variance. Furthermore, we discover that an overlooked collapse pattern, namely dimensional collapse, is indeed reachable when learning representations in a self-supervised fashion. We connect dimensional collapse with strong correlations between axes and consider this connection a strong motivation for feature decorrelation (*i.e.* standardizing the covariance matrix).

Through this work, we hope not only to present to our community the insights regarding the importance and the potential of feature decorrelation but also to facilitate future work that advances self-supervised learning by addressing design flaws instead of mostly trial and error.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *ICLR*, 2020. 1, 2

[2] Mehdi Azabou, Mohammad Gheshlaghi Azar, Ran Liu, Chi-Heng Lin, Erik C. Johnson, Kiran Bhaskaran-Nair, Max Dabagia, Keith B. Hengen, William R. Gray Roncal, Michal Valko, and Eva L. Dyer. Mine your own view: Self-supervised learning through across-sample prediction. *CoRR*, abs/2102.10106, 2021. 1

[3] Anthony J. Bell and Terrence J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997. 4

[4] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 8

[5] 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. 1, 2

[6] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. 1, 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020. 1, 2, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2, 6, 7, 8

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 8

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 2, 6, 7

[11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[12] Michael Cogswell, Faruk Ahmed, Ross B. Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 1

[14] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. *arXiv preprint arXiv:2007.06346*, 2020. 2, 6

[15] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with "bootstrap your own latent" (byol). https://generallyintelligent.ai/, 2020. 6

[16] Divya Gaur, Joachim Folz, and Andreas Dengel. Training deep neural networks without batch normalization. *arXiv preprint arXiv:2008.07970*, 2020. 6

[17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[18] 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. 2, 6

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 6

[21] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 2

[22] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *ICLR*, 2019. 2

[23] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[24] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. *arXiv preprint arXiv:2011.08435*, 2020. 1, 8

[25] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018. 2, 3, 4

[26] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4874–4883, 2019. 2

[27] Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *CoRR*, abs/2011.11765, 2020. 1

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 2, 3

[29] Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 8

[30] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6

[31] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015. 6

[32] Chunyuan Li, Xiujun Li, Lei Zhang, Baolin Peng, Mingyuan Zhou, and Jianfeng Gao. Self-supervised pre-training with hard examples improves visual representations. *CoRR*, abs/2012.13493, 2020. 1

[33] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 8

[34] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2, 8

[36] Colorado Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning, 2020. 1

[37] Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020. 2, 7

[38] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *CoRR*, abs/2010.04592, 2020. 1

[39] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[40] 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. 6

[41] Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and coloring batch transform for gans. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2

[42] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021. 2

[43] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 1

[44] Feng Wang, Huaping Liu, Di Guo, and Fuchun Sun. Unsupervised representation learning by invariance propagation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[45] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 2020. 2

[46] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018. 2, 8

[47] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *CoRR*, abs/1610.05256, 2016. 1

[48] Wei Xiong, Bo Du, Lefei Zhang, Ruimin Hu, and Dacheng Tao. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, editors, *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 519–528. IEEE Computer Society, 2016. 2

[49] Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Hongkai Xiong, and Qi Tian. Seed the views: Hierarchical semantic alignment for contrastive representation learning, 2021. 1

[50] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 6

[51] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 2, 7

[52] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019. 8