Align Representations with Base: A New Approach to Self-Supervised Learning

Shaofeng Zhang[1], Lyn Qiu[1], Feng Zhu[2], Junchi Yan[1]*, Hengrui Zhang[1],
Rui Zhao[1,2], Hongyang Li[2], Xiaokang Yang[1]
[1]Shanghai Jiao Tong University, [2]SenseTime Research
{sherrylone, lyn_qiu, yanjunchi, sqstardust, xkyang}@sjtu.edu.cn
{zhufeng, zhaorui, lihongyang}@sensetime.com

Abstract

*Existing symmetric contrastive learning methods suffer from collapses (complete and dimensional) or quadratic complexity of objectives. Departure from these methods which maximize mutual information of two generated views, along either instance or feature dimension, the proposed paradigm introduces intermediate variables at the feature level, and maximizes the consistency between variables and representations of each view. Specifically, the proposed intermediate variables are the nearest group of base vectors to representations. Hence, we call the proposed method **ARB** (Align Representations with Base). Compared with other symmetric approaches, ARB 1) does not require negative pairs, which leads the complexity of the overall objective function is in linear order, 2) reduces feature redundancy, increasing the information density of training samples, 3) is more robust to output dimension size, which outperforms previous feature-wise arts over **28%** Top-1 accuracy on ImageNet-100 under low-dimension settings.*

1. Introduction

One major bottleneck in deep learning is the scarcity of labeled data, and much attention has been paid to unsupervised learning [15, 17, 30, 33] and self-supervised learning [4, 12, 14, 16, 25, 40]. Among the mainstream approaches, most fall into one of three classes: generative, pretext-tasks-based, and contrastive methods. Generative based methods [15, 30, 39] mainly use pixel-level reconstruction to learn the backbone. However, the backbone usually learns a semantic feature, so it's unnecessary to record the information of each pixel. Therefore, more attempts about discriminative approaches are proposed to train encoders by providing pretext tasks [10, 17, 31], which gain obviously better performance, such as rotation angle [10] and spot artifacts [39]. Among these, contrastive-based methods [4, 5, 19] are the mainstream of current research. The incentive is that views of the same images generated by random augmentation retain similar semantic information. Hence, aligning the two-view embeddings is the key to success. However, directly aligning the embeddings usually causes degenerated solutions [35], which means different samples are mapped to the same points in feature space, as shown on the left side of Fig. 1. This is often due to the lack of proper **objective functions** or **architectures** [6, 32].

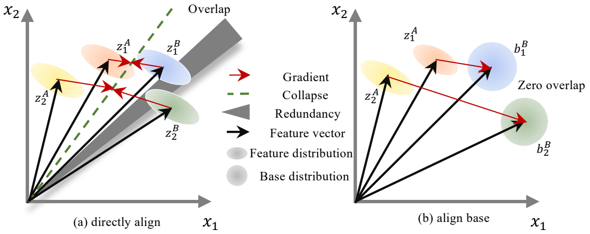

Figure 1. **Comparison of direct alignment (a) and the proposed ARB (b).** z_1^A and z_2^A mean the first and second dimensional feature of view A. Direct alignment will easily cause dimensional collapse (green dash line) and redundancy (gray area). In **ARB**, we solve these caveats by introducing intermediate variables—nearest group of base (b_1^A and b_2^B) to representations. Then, the redundancy reduces to zero without dimensional collapse.

Therefore, one way is to design a suitable objective function. SimCLR [4] takes each two embeddings as one pair, where positive pairs are views of the same images and negative pairs are composed of views of the different images. By expanding the consistency between positive pairs and the difference between negative pairs, models can avoid mapping different samples to the same points. However, such a strategy calcuates the similarity of every two samples, and this brings quadratic complexity. Another way is to build a proper architecture: BYOL [16] and SimSiam [6] propose asymmetric structures, e.g., Stop-Gradient to avoid negative sampling. Despite their promising results and linear com-

[*] Junchi Yan is the correspondence author. Shaofeng Zhang, Lyn Qiu, Junchi Yan, Xiaokang Yang are also with MoE Key Lab of Artificial Intelligence, Artificial Intelligence Institute, Shanghai Jiao Tong University. Rui Zhao is also with Qing Yuan Research Institute, Shanghai Jiao Tong University.

plexity (since no pair-wise distance is required), the rationales behind these approaches are still unclear[1].

To this end, we propose a novel method named ARB to fill the gap. In detail, we maximize the mutual information between the immediate variable generated from one view and representations of the other view, and give a theoretical explanation about why it could avoid degenerate solutions. Technically, we propose to shuffle the feature and divide the output space into several groups on feature dimension to further reduce the complexity. In a nutshell, the highlights of this paper are summarized as:

1) We propose a new method named ARB to avoid collapses in contrastive learning, which is straightforward, comprehensible and efficient (shuffle and divide groups). Compared with other symmetric architecture, ARB only requires linear order complexity of objective (negative-free).

2) We theoretically analyze the relationship between the proposed ARB (maximize the mutual information between the proposed immediate variables and representations) and previous feature-wise methods [37] (maximize the consistency of two views). Furthermore, we give a theoretical analysis of how ARB could avoid degenerate solutions.

3) The experiments results on CIFAR-10, CIFAR-100, and ImageNet show that our method can achieve higher or be on par with previous methods. Compared with other feature-wise contrastive methods like Barlow Twins [1,37], our methods are much more robust to dimension size.

2. Related Works

Since our method directly aligns embeddings with the closest group of base vectors, which is a negative-free method, we briefly introduce the previous methods from the perspective of whether negative samples are involved.

Negative-requiring methods. InfoNCE based methods [4, 19] usually require a large number of negative pairs to boost the accuracy, which is hard to store. Hence, Moco [19] proposes a memory bank module to address this issue. Further, they propose two tricks to prevent collapses, i.e., Stop-Gradient and asymmetrically update encoders [20]. SimCLR [4] proposes a simple yet effective framework to learn representations, where different samples in one mini-batch are regarded as negative pairs. Hence, large batch size is required to boost the accuracy. Inspired by SimCLR, Moco V2 [5] uses stronger augmentation functions to increase the variance of views, which achieves higher accuracy against SimCLR. Besides, work [35] theoretically analyzes the components of InfoNCE. They modify InfoNCE by trading off the alignment part and uniformity part and find the key success for contrastive learning is the alignment part. Then, lots of works focus on generating, mining hard negative pairs [22,24,29,34] and hard positive pairs [11,18] to boost the accuracy. The above work is along the instance dimension, and Barlow Twins [37] first calculate pair-wise correlation on feature dimension across two views, where the pairs composed of the same features across samples of two views are encouraged to align, while the pairs composed of the different features are forced to minimize to 0. VICReg [1] proposes to add instance-wise (variance) regularization on the basis of Barlow Twins [37], which achieves similar accuracy with Barlow Twins.

Negative-free methods. Since alignment is the key to contrastive learning [35], one of the exploring directions is discarding negative pairs, as firstly explored in BYOL [16]. They propose a predictor module and adopt EMA [28] algorithm, stopping gradient to update encoders. SimSiam [6] explores the key to avoiding collapses in the asymmetric architecture empirically and finds the stop-gradient and predictor to be the answer. The work [32] replaces the encoder in BYOL with a two-layer model and gives a theoretical analysis of why the two models (online and target) do not collapse. However, it still remains an open problem why they could learn informative representations. Inspired by classical whitening transformation, e.g., ZCA whitening [21], the work [13] first transforms the learned embeddings before calculating loss. However, the performance is limited due to the inconsistent dimension of whitening (feature-wise) and objective function (instance-wise).

3. Methodology

As a pure negative-free method, ARB is a symmetric thus is more neat and efficient than previous (asymmetric) negative-free methods [6, 16]. We will begin to describe ARB from a previous feature-wise method [37], followed by the framework, objectives, and other used techniques.

3.1. Preliminaries

Self-supervised Learning via Feature Decorrelation. Departure from previous contrastive methods via instance discrimination [4, 16, 19], decorrelation-based methods learn representations via feature-level regularization [1, 37, 38], i.e., maximizing the correlation of the same feature dimension of image representations from two augmented views, and at the same time minimizing the correlation of different feature dimensions. One typical loss for this target refers to the Barlow Twins (BT) loss [37]:

$$\mathbf{L}_{BT} = \sum_i (1 - \mathbf{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathbf{C}_{ij}^2 \tag{1}$$

where $\mathbf{C} = \mathbf{Z^A}^\top \mathbf{Z}^B$ is the cross-correlation matrix between the representations of two views, and $\mathbf{Z}^A$ and $\mathbf{Z}^B \in \mathbb{R}^{N \times d}$ are the column-standard-scaled embedding (0-mean and $1/\sqrt{N}$-standard deviation). N is the batch size and

[1] Note [32] provides an analysis of their learning dynamics with two-layer models, which accounts for the reason why the two models do not fail with trivial solutions. Yet it still remains an open problem why they could learn informative representations.

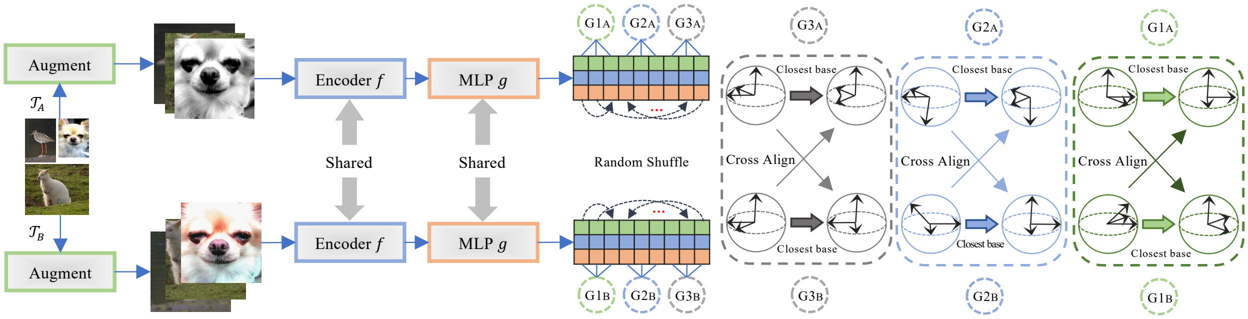

Figure 2. **Framework of ARB**: two augmentations in different distributions. Suppose there are only three samples in one mini-batch, both encoder f and MLP g are weight-sharing. After obtaining embedding matrix $\mathbf{Z}^A$ and $\mathbf{Z}^B$ (each row represents embedding of a sample), we first randomly shuffle the feature dimensions in each mini-batch, then divide the embedding matrix into p groups (three groups in the figure) on the feature dimension, and finally calculate $\mathcal{L}_{ARB}$ in each group and the overall loss is the summation of all the (three) groups.

λ is the hyper-parameter. The loss above has a $\mathcal{O}(d^2)$ time/memory complexity, and usually a large feature dimension d is required for downstream tasks [37].

3.2. The proposed ARB

3.2.1 Nearest Orthonormal Basis

In linear algebra, the orthonormal (basis) matrix of a $d-$dimensional space is a $d \times d$ square matrix $\mathbf{B}_o = [\mathbf{b}_1, \cdots \mathbf{b}_d]$ whose vector are all unit vectors and orthogonal to each other (i.e, $\mathbf{B}_o^\top \mathbf{B}_o = \mathbf{I}$). We extend this concept to non-square cases: we call $\mathbf{B}_o \in \mathbb{R}^{N \times d}$ an orthonormal matrix as long as $\mathbf{B}_o^\top \mathbf{B}_o = \mathbf{I} \in \mathbb{R}^{d \times d}$. Given a standard-scaled embedding matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$ (e.g. the embedding matrix), we then define its nearest orthonormal basis.

Definition 1 *(Nearest Orthonormal Basis, NOB). The nearest orthonormal basis matrix of a standard-scaled matrix* $\mathbf{Z}$ *(termed as* $\mathcal{M}(\mathbf{Z})$*) has the minimum* l_2 *distance to the input matrix, formally:*

$$\mathcal{M}(\mathbf{Z}) = \min_{\mathbf{B}_o} \|\mathbf{Z} - \mathbf{B}_o\|_2^2 \quad s.t. \quad \mathbf{B}_o^\top \mathbf{B}_o = \mathbf{I} \tag{2}$$

When $\mathbf{Z}$ is full column rank (i.e., rank$(\mathbf{Z}) = d$), Eq. 2 has its close-form solution:

Theorem 1 *The optimal solution of equation (Eq. 2) is* $\mathcal{M}(\mathbf{Z}) = \mathbf{Z}\mathbf{\Sigma}^{-\frac{1}{2}} = \mathbf{Z}\mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{U}^\top$*, where* $\mathbf{\Sigma}$ *is the correlation matrix of input variables* $\mathbf{Z}$*, i.e.,* $\mathbf{\Sigma} = \mathbf{Z}^\top\mathbf{Z}$*.* $\mathbf{U}$ *is the eigenvector matrix, and* $\mathbf{\Lambda}$ *is the diagonal eigenvalue matrix of* $\mathbf{\Sigma}$ *respectively (*$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$*).*

Proof 1 *Unfold Eq. 2, we have:*

$$\mathcal{M}(\mathbf{Z}) = \min_{\mathbf{B}_o} \left[tr(\mathbf{Z}^\top\mathbf{Z}) - 2 \cdot tr(\mathbf{Z}^\top\mathbf{B}_o) + tr(\mathbf{B}_o^\top\mathbf{B}_o)\right] \tag{3}$$

As $\mathbf{Z}$ *is standard-scaled, and* $\mathbf{B}_o^\top\mathbf{B}_o = \mathbf{I}$*, we have* $tr(\mathbf{Z}^\top\mathbf{Z}) = tr(\mathbf{B}_o^\top\mathbf{B}_o) = d =$ constant*. Hence, the minimization problem can be transformed to:*

$$\mathcal{M}(\mathbf{Z}) = \max_{\mathbf{B}_o} tr(\mathbf{Z}^\top\mathbf{B}_o) \quad s.t. \quad \mathbf{B}_o^\top\mathbf{B}_o = \mathbf{I} \tag{4}$$

Note that $\mathbf{Z}$ *is a full-rank square matrix and* $\mathbf{\Sigma}$ *is its correlation matrix, so we can find another matrix* $\mathbf{R} \in \mathbb{R}^{d \times d}$ *s. t.* $(\mathbf{Z}\mathbf{R})^\top(\mathbf{Z}\mathbf{R}) = \mathbf{I}$*, where* $\mathbf{R}\mathbf{R}^\top = \mathbf{\Sigma}^{-1}$*. Thus,* $\mathbf{Z}\mathbf{R}$ *is also an orthonormal matrix of the* d*-dimension space, which indicates that there exists another orthonormal matrix* $\mathbf{T} \in \mathbb{R}^{d \times d}$*, s. t.* $\mathbf{B}_o\mathbf{T} = \mathbf{Z}\mathbf{R}$*. Then, the target becomes:*

$$\max_{T} tr(\mathbf{Z}^\top\mathbf{Z}\mathbf{R}\mathbf{T}^{-1}) \quad s.t. \quad \mathbf{T}^\top\mathbf{T} = \mathbf{I} \tag{5}$$

where $\mathbf{R} = \mathbf{\Sigma}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top$*, so we have:*

$$\begin{aligned} tr(\mathbf{Z}^\top\mathbf{Z}\mathbf{R}\mathbf{T}^{-1}) &= tr\ (\mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{U}^\top\mathbf{T}^{-1}) \\ &= \sum_i \mathbf{\Lambda}_{ii}^{1/2}(\mathbf{U}^\top\mathbf{T}^{-1}\mathbf{U})_{ii} \end{aligned} \tag{6}$$

Since $\mathbf{T}^{-1}$ *and* $\mathbf{U}$ *are both orthonormal,* $\mathbf{U}^\top\mathbf{T}^{-1}\mathbf{U}$ *is also an orthonormal matrix. So we have* $(\mathbf{U}^\top\mathbf{T}^{-1}\mathbf{U})_{ii} \leq 1$*. Note that* $\mathbf{U}$ *is a rotation matrix, i.e.,* $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$*. Thus, we can obtain the maximum of Eq. 6 if and only if* $\mathbf{T}^{-1} = \mathbf{I}$*.*

Take $\mathbf{T}^{-1} = \mathbf{I}$ *, note that* $\mathbf{B}_o = \mathbf{Z}\mathbf{R}\mathbf{T}^{-1}$ *and* $\mathbf{R} = \mathbf{\Sigma}^{-1/2}$*, then we have* $\mathcal{M}(\mathbf{Z}) = \mathbf{Z}\mathbf{\Sigma}^{-1/2} = \mathbf{Z}\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top$*, from this we complete the proof.* □

Theorem 1 indicates that for a given full column rank $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^N$, where $\mathbf{z}_i \in \mathbb{R}^{1 \times d}$, one can always find its nearest orthonormal base by $\mathcal{M}(\mathbf{Z})$.

3.2.2 Align Representations with NOBs

Given a batch of input images $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, we first generate two views of the input data through random augmentation(transformation) [37], and term the two views as $\mathbf{X}^A$ and $\mathbf{X}^B$ respectively. Then, we feed $\mathbf{X}^A$ and $\mathbf{X}^B$ to a shared encoder $f_\theta(\cdot)$ followed by a projector $g_\gamma(\cdot)$ [4, 19] to get representations $\mathbf{H}$ and outputs respectively: $\mathbf{H} = f_\theta(\mathbf{X})$, $\mathbf{Z} = g_\gamma(\mathbf{H}) \in \mathbb{R}^{N \times d}$. Following [1, 37, 38], the output $\mathbf{Z}$ is further standardized along the batch dimension:

$$\mathbf{Z}_{:,i} = \frac{\mathbf{Z}_{:,i} - \mu_i}{\sqrt{N}\sigma_i}, \quad i = 1, \cdots, d \tag{7}$$

where μ_i and σ_i are the mean and standard deviation of the i-th dimension of $\mathbf{Z}$ respectively. Next, we calculate the nearest orthonormal bases of the output matrices of the two views by $\mathbf{B}^A = \mathcal{M}(\mathbf{Z}^A)$ and $\mathbf{B}^B = \mathcal{M}(\mathbf{Z}^B)$ as described in Eq. 2. Finally, we learn representations through minimizing the distance between the output matrix of one view and the nearest orthonormal basis matrix of the other view:

$$\mathcal{L}_{ARB} = tr(\|\mathbf{I} - (\mathbf{Z}^A)^\top \mathbf{B}^B\|_2^2) + tr(\|\mathbf{I} - (\mathbf{Z}^B)^\top \mathbf{B}^A\|_2^2) \tag{8}$$

Note that our objective function Eq. 8 is essentially maximizing the similarity between the output of a view and the orthonormal base of another view, so we call our method **Aligning Representation with Base (ARB)** in this paper.

3.2.3 Towards implementations on real-world data

Non-full rank cases. In most cases, samples are fed to models by a random mini-batch and the batch size K is smaller than output dimensions d. In this condition, we can't get the closest base from $\mathcal{M}$, since $\mathbf{\Sigma}$ may not be reversible and we can not find d orthogonal vectors in K-dimensional space. Thus, a more general solution is to find a pseudo base [7]. Given an embedding matrix of one mini-batch samples $\mathbf{Z} \in \mathbb{R}^{N \times d}$ ($d < N$), we calculate the correlation matrix by $\mathbf{\Sigma} = \mathbf{Z}^\top \mathbf{Z}$. Then, the pseudo base is:

$$\mathbf{B}_{pseudo} = \mathbf{Z}\left[\mathbf{U}^\top (\mathbf{\Lambda} + \lambda \mathbf{I})^{-\frac{1}{2}} \mathbf{U}\right] \tag{9}$$

where λ is a hyper-parameter (10^{-4} by default), and $\mathbf{\Sigma} = \mathbf{U}^\top \mathbf{\Lambda} \mathbf{U}$ is spectral decomposition of the correlation matrix.

3.2.4 Reduce computation complexity

As presented above, calculating the nearest base or pseudo base requires matrix decomposition, which is cubic with respective to the output dimension. Although this process does not require back-propagation, it is still time-consuming. To further reduce the complexity, we split the d-dimension feature into p groups. For each group in d/p-dimension space, we find its closest base through matrix decomposition. However, when p is large, the strategy will lose much information, i.e., features in the same group are orthogonal, while features in different groups are non-orthogonal. As such, the redundancy will remain in features in different groups, i.e., the redundancy increases with the increase of the number of groups. Hence, we propose a new technology to alleviate this phenomenon, i.e., shuffle the feature dimension before grouping. Consider loss objective function of Barlow Twins [37], the cross-correlation matrix takes $\mathcal{O}(N \cdot d^2)$ FLOP. Then, element-wise MSE loss is calculated, which takes about $\mathcal{O}(d^2)$ FLOP. Further, backpropagation takes about $\mathcal{O}(N \cdot d^2)$ FLOP. In ARB, by rotating the base $\mathbf{B}_\omega$, it totally takes about $\mathcal{O}(N \cdot d^2 + d^3 + N^2 \cdot d)$ FLOP, where $N \approx d$. After applying group-wise alignment, the complexity is further reduced to $\mathcal{O}(p \cdot (N \cdot (d/p)^2 + (d/p)^3 + N^2 \cdot (d/p)))$, where $\frac{d}{p} << d$, especially for large output dimension d. In experiments, we find that with the random-shuffle trick, ARB improves about 4% $\sim$ 6% top-1 accuracy.

3.3. Intuitive analysis

It has been shown that current instance-level contrastive learning methods [6, 16] often outperform feature-level methods [1, 37]. Now we explain why we insist on feature dimension rather than instance dimension, mainly in two aspects. **1) Size of dimension and mini-batch.** Following [35], the alignment term of InfoNCE [31] (widely used in instance-level methods [4, 19]) can be formulated as:

$$\mathbb{E}_{(x,x^+)\sim p_{pos}}[-f(x)^\top f(x^+)/\tau] \geq \mathbb{E}_{(x,x^+)\sim p_{pos}}[-\frac{1}{\tau}] \tag{10}$$

The above inequality gets equal if and only if $f(x) = f(x^+)$. While the uniformity part [35] pulls all the embeddings in a hyper-sphere uniformly. We know that N data points can not be uniformly distributed on space below $N+1$ dimensions. Otherwise, the feature values of the remaining dimensions will stay the same (dimension collapse [27]), forcing them to be uniformly distributed. However, in ARB, we usually set $d \geq N$, which is unsuitable to apply on instance dimension. **2) Hard positives [11].** The proposed ARB always tries to decorrelate two vectors to orthogonal, while some different instances should be close to each other, i.e., hard positives. Such property of ARB may harm the accuracy when apply on instance dimension. We also try applying ARB on instance dimension, which we will discuss in experiments (see ARB(ins-fea) in Table 5).

3.4. Theoretical analysis

Theorem 2 *The proposed $\mathcal{L}_{ARB}$ in Eq. 8 is the upper bound of the invariance term in $\mathcal{L}_{BT}$ in Eq. 1. Besides, minimizing $\mathcal{L}_{ARB}$ equals to minimizing both invariance and decorrelation terms in $\mathcal{L}_{BT}$.*

Proof 2 *Recall the first term of $\mathcal{L}_{ARB}$ is $tr(\|\mathbf{I} - (\mathbf{Z}^A)^\top \mathbf{B}^B\|_2)$, we have:*

$$tr((\mathbf{Z}^A)^\top \mathbf{B}^B) = tr((\mathbf{Z}^A)^\top \mathbf{Z}^B \mathbf{U}^\top \mathbf{\Lambda}_C^{-\frac{1}{2}} \mathbf{U}) \tag{11}$$

Denote the cross correlation matrix as Σ_{C1}, then, we have:

$$\begin{aligned} tr((\mathbf{Z}^A)^\top \mathbf{B}^B) &= tr(\mathbf{V}^\top \mathbf{\Lambda}_{C1} \mathbf{V} \mathbf{U}^\top \mathbf{\Lambda}_C^{-\frac{1}{2}} \mathbf{U}) \\ &\leq tr(\mathbf{U}^\top \mathbf{\Lambda}_{C1} \mathbf{\Lambda}_C^{-\frac{1}{2}} \mathbf{U}) \end{aligned} \tag{12}$$

where $\mathbf{V}$ and $\mathbf{\Lambda}_{C1}$ are eigenvectors and eigenvalues of Σ_{C1}. Since $\mathbf{Z}^A$ and $\mathbf{Z}^B$ are both centered vectors, we have $(\mathbf{\Lambda}_{C1})_{ii} < (\mathbf{\Lambda}_C)_{ii}$, i.e., the proposed $\mathcal{L}_{ARB}$ is the upper bound of the invariance term in $\mathcal{L}_{BT}$. Then, since $\mathbf{B}^B$ is *orthogonal, by optimizing $\mathcal{L}_{ARB}$, $(\mathbf{\Sigma}_{C1})_{ij}$ will be reduced to 0, which is consistent with the decorrelation term in $\mathcal{L}_{BT}$. Then, we can complete the proof.* □

The theorem shows the relation between $\mathcal{L}_{ARB}$ and $\mathcal{L}_{BT}$, where $\mathcal{L}_{ARB}$ is the upper bound of the invariance term in $\mathcal{L}_{BT}$. Further, by maximizing the consistency between $\mathbf{Z}^A$ and $\mathbf{B}^B$, we can directly discard the decorrelation term of $\mathcal{L}_{BT}$, in a linear complexity for objective function.

Theorem 3 *The mutual information (MI) $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B)$ is the upper bound of $\mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$. Besides, maximizing the MI of $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B)$ is equivalent to maximizing $\mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$:*

$$\max_{f,g} \mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B) = \max_{f,g} \mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B) \tag{13}$$

where $\mathcal{Z}$ and $\mathcal{B}$ are the variable of embeddings and basis.

Proof 3 *The proof is based on:*

$$\begin{aligned} \mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B) &= \mathcal{H}(\mathcal{B}^B) - \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^A) \\ \mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B) &= \mathcal{H}(\mathcal{Z}^B) - \mathcal{H}(\mathcal{Z}^B|\mathcal{Z}^A) \end{aligned} \tag{14}$$

where $\mathcal{H}(\mathcal{B}^B) \geq \mathcal{H}(\mathcal{Z}^B)$. Consider in full-rank condition, i.e., $\lambda = 0$ and $\mathcal{H}(\mathcal{Z}^B|\mathcal{B}^B) = \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^B) = 0$, we have $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B) \geq \mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$. Besides, since $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B) = \mathcal{H}(\mathcal{Z}^A) = \mathcal{H}(\mathcal{B}^B)$, maximizing the MI of $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B)$ is equivalent to maximizing $\mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$. Then, consider in non full-rank condition. We know $\mathcal{H}(\mathcal{B}^B|\mathcal{Z}^B) = 0$, i.e., $\mathcal{Z}^B$ to $\mathcal{B}^B$ is injective, while the inversion does not hold (due to the bias λ). Hence, we have $\mathcal{H}(\mathcal{Z}^B) \geq \mathcal{H}(\mathcal{Z}^B|\mathcal{B}^B) \geq \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^B) = 0$. By the chain rule, we have:

$$\begin{aligned} \mathcal{H}(\mathcal{B}^B, \mathcal{Z}^B|\mathcal{Z}^A) &= \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^B, \mathcal{Z}^A) + \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^A) \\ &= \mathcal{H}(\mathcal{Z}^B|\mathcal{B}^B, \mathcal{Z}^A) + \mathcal{H}(\mathcal{Z}^B|\mathcal{Z}^A) \end{aligned} \tag{15}$$

where $\mathcal{H}(\mathcal{B}^B|\mathcal{Z}^B, \mathcal{Z}^A) \leq \mathcal{H}(\mathcal{Z}^B|\mathcal{B}^B, \mathcal{Z}^A)$. Thus, we have $\mathcal{H}(\mathcal{Z}^B|\mathcal{Z}^A) \geq \mathcal{H}(\mathcal{B}^B|\mathcal{Z}^A)$. Take the inequality to Eq. 14, $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B) \geq \mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$ holds. □

The above theorem indicates that when maximizing the mutual information between $\mathcal{Z}^A$ and $\mathcal{B}^B$, we can also maximize the mutual information between $\mathcal{Z}^A$ and $\mathcal{Z}^B$.

Theorem 4 *Define the consistency between two groups of vectors as $con(\mathbf{Z}, \mathbf{H}) = tr(\mathbf{Z}^\top \mathbf{H})$. Then, λ is the factor to influence the distance between input data and pseudo base. Besides, λ is smaller, the consistency $con(\mathbf{Z}, \mathbf{H})$ is larger.*

The proof is given in supplementary. Theorem 4 indicates with smaller λ, the distance between the pseudo base and original data will be smaller. Correspondingly, the estimated mutual information between $\mathcal{I}(\mathcal{Z}^A, \mathcal{B}^B)$ is more accurate to $\mathcal{I}(\mathcal{Z}^A, \mathcal{Z}^B)$ (gets equals when $\mathbf{Z}^B$ is full-rank), which is consistent with our experiments (see Fig. 6). Next, we analyze the maximum consistency between representations and pseudo basis ($d > N$) $con(\mathbf{Z}, \mathbf{B}_{pseudo})$. Since $d > N$, there are at least $d - N$ eigenvalues equal to 0 and the maximum value of $con(\mathbf{B}_{pseudo}, \mathbf{Z})$ is obtained when the column rank of $\mathbf{Z}$ equals to N. Then the maximum value is $(\frac{d \cdot N}{\sqrt{d} + N \cdot \lambda})^{-1/2}$, which is obtained when all the non-zero eigenvalues equal to $\sqrt{d}/N$.

One of the questions is still unclear, which is why directly minimizing $\mathcal{L}_{ARB}$ can avoid collapse. Usually, there are two kinds of collapses, where complete collapse means the model maps all the input data to the same points in hypersphere and dimensional collapse means the data points are not projected onto the hypersphere, but they distribute nearly as a line in the space, making them hard to discriminate. We further give the following theoretical result.

Theorem 5 *By optimizing $\mathcal{L}_{ARB}$, model can avoid collapses (both complete and dimensional collapses).*

Proof 4 *Consider the embeddings of mini-batch data, note that the dimensional collapse is the upper bound of complete collapse and if we avoid dimensional collapse, we can avoid complete collapse. In the worst case, consider we have already gotten the degenerate solution, i.e. $\mathbf{z}_i = \mathbf{z}_j$, where $\mathbf{z}_i$ is the embedding of sample i. By $\mathcal{L}_{ARB}$, we have:*

$$\frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{i,:}} = 2 \cdot (1 - (\mathbf{Z}^A_{i,:})^\top \mathbf{B}^B_{i,:}) \cdot \mathbf{B}^B_{i,:} \tag{16}$$

where $\mathbf{Z}^A_{i,:}$ is the vector composed by i-th feature values in view A and $\frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{i,:}}$ is the partial derivative of $\mathbf{Z}_{i,:}$. We have:

$$\left(\frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{i,:}}\right)^\top \frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{j,:}} = 0, \ \left\|\frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{i,:}}\right\|_2 + \left\|\frac{\partial \mathcal{L}_{ARB}}{\partial \mathbf{Z}_{j,:}}\right\|_2 \neq 0 \tag{17}$$

The above formulates are according to that even if $(\mathbf{Z}^A_{i,:})^\top \mathbf{B}^B_{i,:} = 0$, $(\mathbf{Z}^A_{j,:})^\top \mathbf{B}^B_{j,:}$ must be non-zero value since $\mathbf{B}^B_{j,:} \perp \mathbf{B}^B_{i,:}$. Then, by optimizing in one step, we avoid the collapses. In a more general condition, the two vectors $(\mathbf{Z}^A_{i,:})^\top$ and $(\mathbf{Z}^A_{j,:})^\top$ are optimized in the vertical direction. Thus, they can only intersect in one point in hypersphere and if the intersection point is not the origin, we can avoid collapse perfectly. □

4. Experiments

4.1. Experiment setup

Datasets. We evaluate the proposed method on following datasets, as commonly used in previous self-supervised methods [4, 6, 37].

1) CIFAR-10 and CIFAR-100 [26], two small-scale datasets for 32 × 32 images with 10 and 100 classes, respectively. 2) ImageNet-100 and ImageNet-1k [9] include

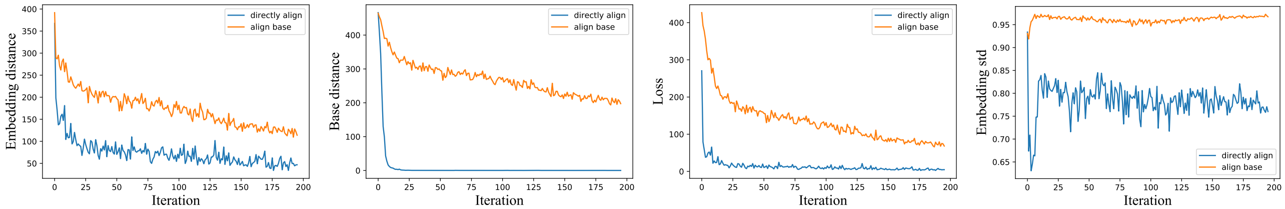

Figure 3. Euclidean distance of embeddings, basis, loss and variance plots over 20k iteration.

Table 1. **Main comparison on CIFAR-10, CIFAR-100 and ImageNet-100**. Proj. and Pred. mean the output dimension in projector and predictor. Negs. means whether to use negative pairs (either feature-wise or instance-wise). M means the number of views.

	Method	Proj. dim #	Pred. dim #	Negs. used ?	Complexity (objective)	CIFAR-10		CIFAR-100		ImageNet-100	
						Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
Asymmetric	BYOL [16]	4096	256	✗	$\mathcal{O}(N)$	92.61	**99.82**	**70.18**	91.36	**80.09**	94.99
	DINO [3]	256	–	✗	$\mathcal{O}(N)$	89.19	99.31	66.38	90.18	74.84	92.92
	SimSiam [6]	2048	512	✗	$\mathcal{O}(N)$	90.51	99.72	65.86	89.48	77.04	94.02
	MoCo V2 [5]	256	–	✓	$\mathcal{O}(NK)$	**92.94**	99.79	69.54	**91.49**	78.2	**95.5**
	ReSSL [41]	256	–	✓	$\mathcal{O}(N^2)$	90.63	99.62	65.83	89.51	76.59	94.41
Symmetric	VICReg [1]	2048	–	✓	$\mathcal{O}(N+d^2)$	90.07	99.71	68.54	90.83	79.22	95.06
	SwAV [2]	256	–	✓	$\mathcal{O}(NC)$	89.17	99.68	64.67	88.52	74.28	92.84
	W-MSE [13]	256	–	✗	$\mathcal{O}(NM^2)$	88.18	99.61	61.29	87.11	69.06	91.22
	SimCLR [4]	256	–	✓	$\mathcal{O}(N^2)$	90.74	99.75	65.39	88.58	77.48	93.42
	Barlow Twins [37]	256	–	✓	$\mathcal{O}(d^2)$	87.39	99.42	57.92	85.23	67.21	90.64
	Barlow Twins [37]	2048	–	✓	$\mathcal{O}(d^2)$	89.57	99.73	69.18	91.19	78.62	94.72
	ARB	256	–	✗	$\mathcal{O}(d)$	91.81	99.86	68.19	91.12	74.86	93.06
	ARB	2048	–	✗	$\mathcal{O}(d)$	**92.19**	**99.89**	**69.57**	**91.77**	**79.48**	**95.51**

100 and 1000 classes, respectively. The datasets are well-balanced in class distribution and the images contain an iconic view of objects, as widely used in vision [20, 37].

Augmentation. Each input image is transformed twice to generate two different views mentioned before. The image augmentation pipeline is listed as follows: random cropping, resizing to 224 × 224 (32 × 32 for CIFAR), horizontal flipping, color jittering, converting to gray-scale, Gaussian blurring, and solarization. The last five are applied randomly on two views with different probabilities, which are exactly the same as [37].

Architecture. Following recent works [4, 37], the encoder adopts ResNet-50 (2048 output units) or ResNet-18 (512 output units) [20] architecture without the final classification layer, followed by an MLP module. The architecture of MLP is the same as [37].

Optimization. Similar to previous works [16, 37], we use the LARS optimizer [36] on all the datasets. We use a learning rate of 0.2 for the weights and 0.005 for the biases and batch normalization parameters. We multiply the learning rate by batch size and divide it by 256. We use a learning rate warm-up period of the first 10 epochs, after which we reduce the learning rate by a factor of 1000 using a cosine decay scheduler [23]. For CIFAR-10 and CIFAR-100, we use single 1080 GPU. For ImageNet-100, the batch size is set as 128 as default, and we use 8 Tesla V100 16G GPUs. For ImageNet-1k, we evaluate ARB on 64 1080Ti GPUs with 256, 2048 and 8192 output dimensions, respectively. The batch size on ImageNet-1k is set 512 as default.

Evaluation. We train a linear classifier on three vision datasets on top of fixed representations of ResNets pre-trained by ARB. Specifically, the linear classifier is trained for 100 epochs with a learning rate of 0.3 and a cosine learning rate scheduler. We minimize the cross-entropy loss with SGD optimizer with momentum 0.9 and weight decay 1e-6. In line with previous arts [4, 37], we set batch size as 256. At the inference stage, we resize the image to 256 × 256 and center crop it to a size of 224 × 224.

4.2. Overall evaluation

Classification task. We mainly divide the contrastive learning methods into two parts, i.e., asymmetric and symmetric architectures. Previous methods [6, 16, 19] with asymmetric architecture achieve state-of-the-art performance against those with symmetric architecture [4, 37], by designing stop gradient and predictor module. However, they suffer from the lack of explainability [32]. Hence, we mainly compare our methods with symmetric methods. Table 1 and 2 give classification results on CIFAR-10, CIFAR-100 and ImageNet datasets with ResNet-18 as

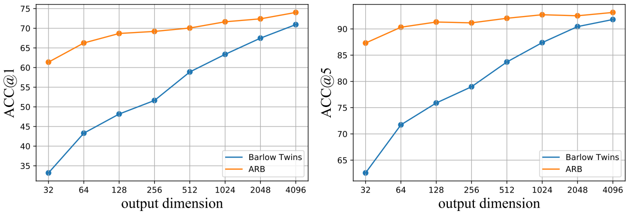

Figure 4. Output dim: ImageNet-100 w/ 100-epoch pre-train.

Table 2. Accuracy on ImageNet with ResNet-50.

Method	Dim	50 eps		100 eps	
		acc@1	acc@5	acc@1	acc@5
Barlow Twins	256	45.38	70.55	52.79	77.48
	2048	52.9	77.92	59.19	82.29
	8192	61.02	84.16	67.74	88.33
ARB	256	49.95	75.42	58.17	80.91
	2048	58.84	81.31	64.42	85.87
	8192	**62.05**	**84.49**	**68.21**	**88.91**

backbone [20]. For CIFAR-10 and CIFAR-100, we set batch size as 256 and train each method with 1,000 epochs. For ImageNet-100, we set batch size as 128 and train each model with 400 epochs. Thanks to the work [8], we can quickly reproduce the results of previous methods. For ImageNet-1k, we pre-train the encoder (ResNet-50 [20]) with 100, 400 epochs with batch size 512. On CIFAR-10, CIFAR-100 and ImageNet-100, ARB achieves the highest accuracy in symmetric methods. Moreover, ARB outperforms baseline Barlow Twins [37] **12.27**% top-1 accuracy with 256 output dimension.

4.3. Ablation study

Align base. We conduct experiments to show the proposed ARB can avoid collapses and report the variance and loss track in Fig. 3. Specifically, we search the closest basis of each view, and report the Euclidean distance between the two bases (second plot in Fig. 3). The track "Directly align" is the designed baseline, which directly aligns embeddings of two views on feature dimension. Then, we train the two methods (directly align and align base) in 1,000 epochs, and report the top-1 and top-5 accuracy under linear evaluation and KNN evaluation in Table 3. As shown in Table 3, although directly aligning embeddings of two views doesn't bring complete collapses, it causes dimensional collapse [38] with poor linear evaluation performance (16.98% top-1 and 41.26% top-5 accuracy), which is consistent with the results in [37]. Recall that the proposed ARB aligns representations with base, which will increase the entropy and variance of $\mathbf{Z}$ (shown in Theorem 2 and right plot of Fig. 3), the top-1 accuracy gets 71.10%.

Output dimension. Since ARB is essentially a feature-wise method (despite the introduced intermediate variables), we conduct robustness test on ImageNet-100. We mainly compare with [37]. The reported results are reproduced from their official code under the same setting. We set the max epoch 100 and batch size 128. The projector dimension is set as 2048-2048-$OutDim$, where $OutDim$ is from 32 $\sim$ 4096. We find that Barlow Twins is heavily influenced by output dimension, while our method is much more robust (Fig. 4). In top-1 accuracy, ARB outperforms Barlow Twins by **28.19**% with 32-dimensional output, and still outperforms Barlow Twins by 3.1% with a rather large output dimension of 4096.

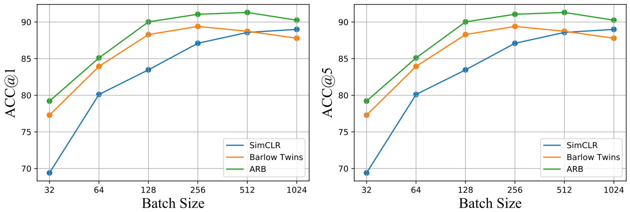

Figure 5. Batch size: CIFAR-100 w/ 50k-iterations pre-train.

Table 3. **Comparison of direct alignment and aligning base**. L means linear evaluation and k in KNN model is set as 5.

Method	L@1	L@5	KNN@1	KNN@5
Directly align	16.98	41.26	1.10	10.20
Align base	**68.19**	**91.12**	**71.10**	**89.60**

Batch size. In line with [4, 37], we test the robustness under small batch sizes. We train all methods with the same 50K **iterations**. Fig. 5 shows the top-1 and top-5 accuracy of SimCLR, Barlow Twins, and our ARB. SimCLR is heavily influenced by batch size, which has also been verified in [4, 6]. Feature-wise methods as Barlow Twins and ARB are more robust to batch size, and our method can get higher accuracy than [37] under all the tested batch sizes.

Groups. As mentioned above, we design shuffle and grouping operation on feature dimension to reduce the complexity, which may also bring negative impact (feature in different groups may be not orthogonal), we conduct extensive experiments on CIFAR-100 dataset to analyze the effect of the number of groups, which are shown in Fig. 6. We set the dimension of projection as 2048-2048-2048. The number of groups and batch size are both set as 256. We find that the accuracy with 8 groups is better than a single group, which may be because, in 256-dimension space, we can not find the 2048 orthogonal vectors. However, if we divide the 2048 dimensions into 8 groups, where each group is a 256-dimension space. We can get unbiased orthogonal vectors (if the embedding matrix $\mathbf{Z}$ is full-rank). We also find with too many groups (256), the accuracy decreases with a large range, which is because features in different groups bring much redundancy. After applying the shuffle operation, the accuracy will drop at a much slower rate.

Convergence rate. We show accuracy curves during training in Fig. 7. The experiments are conducted on CIFAR-100, where we set the max epochs as 100 and use

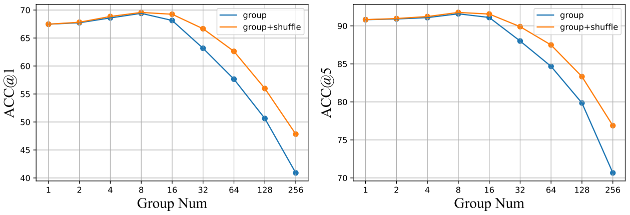

Figure 6. Number of groups: CIFAR-100 w/ 1k-epoch pre-train.

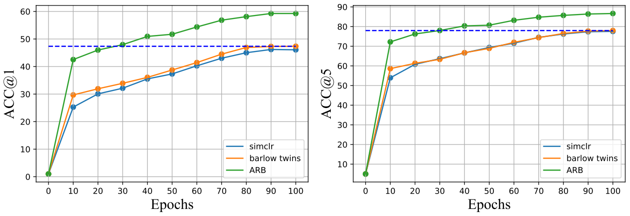

Figure 7. Comparison of convergence rate.

cyclic cosine annealing learning rate scheduler [23], which is commonly used in previous SSL methods [4, 37]. Comparing our method with symmetric methods [4, 37] in every 10 epochs. ARB achieves 47.94% top-1 accuracy at 30 epochs, while the best top-1 accuracy of SimCLR and Barlow Twins (90 and 100 epochs) are 46.08% and 47.35%.

Loss function. We alter our loss in Eq. 8 in several ways to test the necessity of each term (standard scale, batch normalization) and evaluate if the negative samples can improve the accuracy. Experimental results are reported in Table 4. We consider using MSE loss with l_2 normalized embeddings on feature dimension and adding the off-diagonal term in Barlow Twins [37]. We also try removing the Batch Normalization module in MLP. For the off-diagonal term, we first calculate the cross-correlation matrix by $\mathbf{C}_A = (\mathbf{Z}^A)^\top \mathbf{B}^B$, $\mathbf{C}_B = (\mathbf{Z}^B)^\top \mathbf{B}^A$. Then, we alter the ARB loss to Barlow Twins' loss in Eq. 1, and the accuracy is slightly reduced. This proves our method does not require pair-wise decorrelation. For l_2 normalization, we first standardize the embeddings $\mathbf{Z}^A$. Then, we use the function $\mathcal{M}$ to find the closest base, and calculate l_2 normalized embeddings and base, respectively. Finally, we perform MSE loss on the normalized matrices. By normalization, the accuracy is slightly reduced, being consistent with [37].

Instance dimension. Although the optimal solution of instance-wise methods is not orthogonal representations (Sec. 3.3), we also try applying our method on instance dimension. Table 5 gives the accuracy of applying ARB on both instance and feature dimensions, where ARB (fea) gets higher accuracy than ARB (ins) on both CIFAR-10 and CIFAR-100 datasets. One possible reason is that, our ARB applied on instance dimension always tries to align instance representations to instance-wise base, thus requiring all instances to be orthogonal to each other, which may harm the representation learning process when facing **hard positive** pairs that should be as close as possible in the feature space.

Table 4. **Exploration on losses** (ImageNet-100 linear evaluation accuracy with 400-epoch pre-training). BN: batch normalization.

Method	Top-1	Top-5
ARB (standard version)	**79.48**	**95.51**
Add off-diagonal	74.18	92.91
No BN in MLP	79.10	94.69
No BN in MLP w/o normalization	64.18	88.16
MSE loss after ℓ_2 normalization	69.12	91.08

Table 5. **Align on instance or feature dimensions** (linear evaluation accuracy with 1k-epoch pre-training).

Method	Dimension	CIFAR-10		CIFAR-100	
		Top-1	Top-5	Top-1	Top-5
ARB (ins)	256	89.17	99.29	65.59	89.81
ARB (fea)	256	91.81	99.86	68.19	91.12
ARB (ins)	2048	89.31	99.26	65.36	89.56
ARB (fea)	2048	**92.19**	**99.89**	**69.57**	**91.77**

Relation to Barlow Twins [37]. ARB learns representations by aligning embeddings of one view with the searched base of the other view. It can achieve both invariance and decorrelation as in Barlow Twins with linear complexity. Intuitively, ARB only requires alignment, which makes it more robust to output dimension.

Relation to SimSiam [6]. The intuition behind SimSiam is maximizing the consistency between $h(z_i^A)$ and z_i^B, where h indicates the predictor module. Compared with SimSiam, ARB replaces the predictor module to function $\mathcal{M}$. Note that $\mathcal{M}$ has no parameters to optimize, which makes our method more scalable (less storage).

5. Conclusion

We have presented ARB (Align Representations with Base), which aligns the learned embeddings to intermediate variables for self-supervised learning. Compared with previous symmetric methods, ARB does not require pair-wise decorrelation, resulting in a linear order complexity (objective function). We theoretically analyze the relationship between Barlow Twins [37] and ARB, and show why our method can avoid collapses. Besides, we conduct experiments on CIFAR-10, CIFAR-100, and ImageNet. The results show that ARB can achieve higher accuracy than previous symmetric methods [4, 19, 37]. Results of ablations show that ARB is more robust to dimension size than previous methods [4, 37] with a faster convergence rate.

ARB currently can only be used on feature dimension. We hope to extend it on instance dimension. Besides, to find the closest base, we have to calculate the inverse matrix, which is time-consuming and worthy of improvement.

Acknowledgements. This work was in part supported by National Key Research and Development Program of China (2020AAA0107600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and SenseTime Collaborative Research Grant.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv:2105.04906*, 2021. 2, 3, 4, 6

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 6

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 6

[4] 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, 3, 4, 5, 6, 7, 8

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2, 6

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2, 4, 5, 6, 7, 8

[7] Pierre Courrieu. Fast computation of moore-penrose inverse matrices. *arXiv preprint arXiv:0804.4809*, 2008. 4

[8] Victor G Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning. *arXiv preprint arXiv:2108.01775*, 2021. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 1

[11] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *arXiv preprint arXiv:2104.14548*, 2021. 2, 4

[12] Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35:193–234, 2009. 1

[13] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021. 2, 6

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[16] 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. 1, 2, 4, 6

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 1

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020. 2

[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, 3, 4, 6, 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. 2, 6, 7

[21] Nicholas J Higham. *Analysis of the Cholesky decomposition of a semi-definite matrix*. Oxford University Press, 1990. 2

[22] 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. 2

[23] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. 6, 8

[24] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020. 2

[25] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Richard B Larson. Calculations of three-dimensional collapse and fragmentation. *Monthly Notices of the Royal Astronomical Society*, 184(1):69–85, 1978. 4

[28] AJ Lawrance and PAW Lewis. An exponential moving-average sequence and point process (ema1). *Journal of Applied Probability*, 14(1):98–113, 1977. 2

[29] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. I-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*, 2020. 2

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classi-

fication of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 4

[32] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021. 1, 2, 6

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 1

[34] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021. 2

[35] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 1, 2, 4

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

[37] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv:2103.03230*, 2021. 2, 3, 4, 5, 6, 7, 8

[38] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *arXiv preprint arXiv:2106.12484*, 2021. 2, 3, 7

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 1

[40] Shaofeng Zhang, Feng Zhu, Junchi Yan, Rui Zhao, and Xiaokang Yang. Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning. In *International Conference on Learning Representations*, 2021. 1

[41] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *arXiv preprint arXiv:2107.09282*, 2021. 6