Understanding Whitening Loss in Self-Supervised Learning

Lei Huang, Yunhao Ni, Xi Weng, Rao Muhammad Anwer, Salman Khan, Ming-Hsuan Yang, *Fellow, IEEE*, and Fahad Shahbaz Khan

***Abstract*—A desirable objective in self-supervised learning (SSL) is to avoid feature collapse. Whitening loss guarantees collapse avoidance by minimizing the distance between embeddings of positive pairs under the conditioning that the embeddings from different views are whitened. In this paper, we propose a framework with an informative indicator to analyze whitening loss, which provides a clue to demystify several interesting phenomena and a pivoting point connecting to other SSL methods. We show that batch whitening (BW) based methods do not impose whitening constraints on the embedding but only require the embedding to be full-rank. This full-rank constraint is also sufficient to avoid dimensional collapse. We further demonstrate that the stable rank of the embedding is invariant during training by gradient descent, given the assumption that embedding is updated with an infinitely small learning rate. Based on our analysis, we propose channel whitening with random group partition (CW-RGP), which exploits the advantages of BW-based methods in preventing collapse and avoids their disadvantages requiring large batch size. Experimental results on ImageNet classification and COCO object detection reveal that the proposed CW-RGP possesses a promising potential for learning good representations.**

***Index Terms*—Self-supervised learning, whitening, deep neural networks, collapse, stable rank.**

I. Introduction

SELF-SUPERVISED learning (SSL) has made significant progress over the last several years [1], [2], [3], [4], [5], [6], [7], almost reaching the performance of supervised baselines on many downstream tasks [8], [9]. Several recent approaches rely on a joint embedding architecture in which a dual pair of networks are trained to produce similar embeddings for different views of the same image [5]. Such methods aim to learn representations invariant to the transformation of the same input. One main challenge with the joint embedding architectures is how to prevent a *collapse* of representation, in which the two branches ignore the inputs and produce identical and constant output representations [3], [5].

One line of work uses contrastive learning methods that attract different views from the same image (positive pairs) while pulling apart different images (negative pairs), which can prevent constant outputs from the solution space [10]. While the concept is simple, these methods require a large batch size to obtain a good performance [2], [3], [11]. In addition, a negative pair may have the same semantic label [12], which is detrimental to the learning process of such approaches. Another line of work aims to directly match the positive targets without introducing negative pairs. A seminal approach, BYOL [4], shows that an extra predictor and momentum are essential for representation learning. SimSiam [5] further generalizes [4] by empirically showing that stop-gradient is essential for preventing trivial solutions. Recent methods generalize the collapse problem into dimensional collapse [13], [14] (or informational collapse [15]), where the embedding vectors only span a lower-dimensional subspace and would be highly correlated. Thus, the embedding vector dimensions would vary together and contain redundant information. To address the dimensional collapse, a whitening loss is proposed by only minimizing the distance between embeddings of positive pairs under the condition that embeddings from different views are whitened [13], [16]. A typical approach exploiting batch whitening (BW) [17] and imposing the loss on the whitened output [13], [16] has shown to generate promising results.

Although whitening loss has a theoretical guarantee of avoiding collapse, we observe that the empirical results depend on which whitening transformation [18] is used (see Section IV). This interesting observation motivates us to develop effective whitening loss for SSL. In addition, the whitening operation is used to remove the correlation among axes [13], and a whitened representation ensures the examples are scattered in a spherical distribution [16]. As such, one should use the whitened representation for the downstream tasks. However, it is not typically used in practice. To this end, we study the whitening loss and demystify these interesting observations. The main contributions of this work are:

- We decompose the symmetric formulation of whitening loss into two asymmetric losses, where each asymmetric loss requires an online network to match a whitened target. This mechanism provides a pivoting connection to other

Manuscript received 10 August 2023; revised 17 February 2024; accepted 16 June 2024. Date of publication 21 June 2024; date of current version 5 November 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2022ZD0116310, in part by the NSFC under Grant 62106012, and in part by the Fundamental Research Funds for the Central Universities. Recommended for acceptance by M. Cho. *(Corresponding authors: Lei Huang; Fahad Shahbaz Khan.)*

Lei Huang, Yunhao Ni, and Xi Weng are with the SKLCCSE, Institute of Artificial Intelligence, Beihang University, Beijing 100191, China (e-mail: huangleiAI@buaa.edu.cn).

Rao Muhammad Anwer, Salman Khan, and Fahad Shahbaz Khan are with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE (e-mail: fahad.khan@liu.se).

Ming-Hsuan Yang is with the University of California at Merced, Merced, CA 95343 USA, and also with Google, Mountain View, CA 94043 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3417438, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3417438

methods and a way to understand why certain whitening transformation fails to avoid dimensional collapse.
- We characterize the extent of dimensional collapse (whitening) using the rank [19] of a matrix and provide theoretical results showing that BW-based methods do not impose whitening constraints on the embedding, but they only require the embedding to be full-rank. This full-rank constraint is also sufficient to avoid dimensional collapse.
- We further delve into the training dynamics of embedding, and theoretically demonstrate that the stable rank of the embedding is invariant during training by gradient descent, given the assumption that embedding is updated with an infinitely small learning rate.
- We propose channel whitening with random group partition (CW-RGP), which exploits the advantages of BW-based methods in preventing collapse and avoids their disadvantages in requiring large batch sizes. Experimental results on ImageNet classification and COCO object detection show that CW-RGP has promising potential in learning good representation.

This paper is based on and extends our early work [20] in terms of several aspects. First, we delve into the training dynamics of embedding and demonstrate that the stable rank of the embedding is invariant during training by gradient descent, given the assumption that embedding is updated with an infinitely small learning rate. We provide mathematical derivations and theoretical results. Second, we extend the previous channel-wise random group partition into a general framework to partition the group randomly along the batch and channel dimensions. We further analyze how and why the partition affects the effects from the perspective of full-rank constraints on the embedding.

II. Related Work

Contrastive learning: Contrastive methods have been proposed to deal with dimensional collapse by attracting positive samples closer and spreading negative samples apart [10], [21]. In these approaches, negative samples play an important role and must be well designed [1], [22], [23]. One typical mechanism is building a memory bank with a momentum encoder to provide consistent negative samples as proposed in MoCos [2], [24] to achieve better results [2], [25], [26], [27]. Other works include SimCLR [3] addresses that more negative samples in a batch with strong data augmentations perform better. As contrastive learning methods require large batch sizes or memory banks and entail heavy computational load, analyzing whether all negative pairs are necessary for the task is imperative.

Asymmetric architecture: Numerous SSL methods have been developed explicitly [4], [5], [28], [29], [30] without using negative pairs. One typical way to avoid representational collapse is the introduction of asymmetric network architecture. BYOL [4] appends a predictor after the online network and introduces momentum into the target network. SimSiam [5] further simplifies BYOL by removing the momentum mechanism and shows that the stop-gradient to the target network is an alternative approximation to the momentum encoder. On the other hand, an asymmetric pipeline with a self-distillation loss for vision transformers [31] is developed. However, how the asymmetric network avoids collapse without negative pairs remains unclear. As such, stop-gradient [5], [32] are used to analyze the training dynamics [33] to draw connections between asymmetric networks with contrastive learning methods [34], [35]. Our work provides a pivoting connection between asymmetric networks and whitening loss to deal with the collapse problem.

Whitening loss: It has been shown that whitening loss has a theoretical guarantee of avoiding collapse by minimizing the distance of positive pairs under the conditioning that the embeddings from different views are whitened [13], [15], [16], [36]. One way to obtain whitened output is imposing a whitening penalty as regularization on embedding, i.e., soft whitening, by Barlow Twins [36], VICReg [15], and CCA-SSG [37]. Another way is using batch whitening (BW) [17], [38], i.e., hard whitening, by W-MSE [16] and Shuffled-DBN [13]. We propose a different hard whitening method, i.e., channel whitening (CW) that has the same function to ensure all the singular values of transformed outputs are one for avoiding collapse. Compared to BW, CW is numerically more stable and works better when the batch size is small. Furthermore, our CW with random group partition (CW-RGP) can effectively control the extent of constraint on embedding and obtain better performance in practice. We note that ICL [39] also aims to decorrelate instances in a way similar to CW but having several significant differences in technical details. ICL uses "stop-gradient" for the whitening matrix, while CW requires back-propagation through the whitening transformation. In addition, ICL uses extra pre-conditioning on the covariance and whitening matrices, which is essential for ICL to keep numerical stability, while CW does not use extra pre-conditioning and can work well since it encourages the embedding to be full-rank.

Understanding collapse in SSL: There are also works aiming at understanding how dimensional collapse occurs [13], [14] and how it can be avoided by using whitening loss [13]. It is clear that the whitening constraint imposed on the embedding [15], [36] can also avoid the dimensional collapse, and has connection to the contrastive methods [35]. The recent works [40], [41], [42] further discuss how to characterize the magnitude of dimensional collapse, and connect the spectrum of a representation to a power law. Distinct from these works, we theoretically demonstrate that BW-based methods do not impose whitening constraints; instead, they only require the embedding to be full-rank. Furthermore, we investigate the training dynamics of the embedding and theoretically prove that the stable rank of the embedding remains invariant during training through gradient descent, given certain mild assumptions.

III. Preliminaries

Let $\mathbf{x}$ denote the input sampled uniformly from a set of images $\mathbb{D}$, and $\mathbb{T}$ denote the set of data transformations available for augmentation. We consider the Siamese network $f_\theta(\cdot)$ parameterized by θ. It takes as input two randomly augmented views, $\mathbf{x}_1 = \mathcal{T}_1(\mathbf{x})$ and $\mathbf{x}_2 = \mathcal{T}_2(\mathbf{x})$, where $\mathcal{T}_{1,2} \in \mathbb{T}$. The network $f_\theta(\cdot)$ is trained with an objective function that minimizes the distance between embeddings obtained from different views of

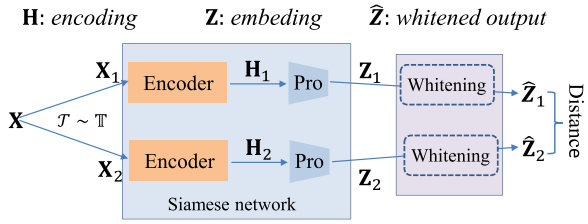


Fig. 1. Basic notations for SSL in this paper. Given the mini-batch inputs $\mathbf{X}$, the encoding, embedding, and whitened output of mini-batch data are denoted by $\mathbf{H}$, $\mathbf{Z}$, and $\widehat{\mathbf{Z}}$, respectively.

the same image:

$$\mathcal{L}(\mathbf{x},\theta)=\mathbb{E}_{\mathbf{x}\sim\mathbb{D},\mathcal{T}_{1,2}\sim\mathbb{T}}\,\ell\left(f_\theta(\mathcal{T}_1(\mathbf{x})),f_\theta(\mathcal{T}_2(\mathbf{x}))\right),\quad(1)$$

where $\ell(\cdot,\cdot)$ is a loss function. A Siamese network usually consists of an encoder $E_{\theta_e}(\cdot)$ and a projector $G_{\theta_g}(\cdot)$. The output $\mathbf{h}=E_{\theta_e}(\mathcal{T}(\mathbf{x}))$ and $\mathbf{z}=G_{\theta_g}(\mathbf{h})$ are referred to as encoding and embedding, respectively. We summarize the notations and use the corresponding capital letters denoting mini-batch data in Fig. 1. As such, we have $f_\theta(\cdot)=G_{\theta_g}(E_{\theta_e}(\cdot))$ with learnable parameters $\theta=\{\theta_e,\theta_g\}$. The encoding $\mathbf{h}$ is usually used as a representation for evaluation by either training a linear classifier [2] or transferring to downstream tasks. This is due to the empirical evidence that $\mathbf{h}$ is shown to obtain significantly better performance than the embedding $\mathbf{z}$ [3], [5].

The mean square error (MSE) of ℓ_2-normalized vectors is usually used as the loss function [5]:

$$\ell(\mathbf{z}_1,\mathbf{z}_2)=\left\|\frac{\mathbf{z}_1}{\|\mathbf{z}_1\|_2}-\frac{\mathbf{z}_2}{\|\mathbf{z}_2\|_2}\right\|_2^2,\quad(2)$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. This loss is also equivalent to the negative cosine similarity, up to a scale of $\frac{1}{2}$ and an optimization irrelevant constant.

Collapse and Whitening Loss: When minimizing (1), a trivial solution known as collapse could occur such that $f_\theta(\mathbf{x})\equiv\mathbf{c}$, $\forall\mathbf{x}\in\mathbb{D}$. The state of collapse will provide no gradients for learning and offer no information for discrimination. Moreover, a weaker collapse condition called dimensional collapse can be easily arrived, for which the projected features collapse into a low-dimensional manifold. As illustrated in [13], dimensional collapse is associated with strong correlations between axes, which motivates the usage of whitening methods to alleviate the dimensional collapse problem. The essential idea of whitening loss [16] is to minimize (1) under the condition that embeddings from different views are whitened, which can be formulated as[1]:

$$\min_\theta\mathcal{L}(\mathbf{x};\theta)=\mathbb{E}_{\mathbf{x}\sim\mathbb{D},\mathcal{T}_{1,2}\sim\mathbb{T}}\,\ell(\mathbf{z}_1,\mathbf{z}_2),$$
$$\text{s.t. }cov(\mathbf{z}_i,\mathbf{z}_i)=\mathbf{I},\ i\in\{1,2\},\quad(3)$$

where $cov(\mathbf{z}_i,\mathbf{z}_i)$ is the covariance of random vector $\mathbf{z}_i$. Whitening loss provides a theoretical guarantee to avoid (dimensional) collapse since the embedding is whitened with all axes decorrelated [13], [16]. While it is difficult to solve the problem of (3) directly, Ermolov et al. [16] propose to whiten the mini-batch embedding $\mathbf{Z}\in\mathbb{R}^{d_z\times m}$ using batch whitening (BW) [17], [43] and impose the loss on the whitened output $\widehat{\mathbf{Z}}\in\mathbb{R}^{d_z\times m}$, given the mini-batch inputs $\mathbf{X}$ with a size of m, as follows:

$$\min_\theta\mathcal{L}(\mathbf{X};\theta)=\mathbb{E}_{\mathbf{X}\sim\mathbb{D},\mathcal{T}_{1,2}\sim\mathbb{T}}\,\|\widehat{\mathbf{Z}}_1-\widehat{\mathbf{Z}}_2\|_F^2$$
$$\widehat{\mathbf{Z}}_i=\Phi(\mathbf{Z}_i),\ i\in\{1,2\},\quad(4)$$

where $\Phi(\cdot)$ denotes the whitening transformation over a mini-batch of data.

Whitening Transformations: As shown in [17], [18], there are infinite possible whitening matrices since any whitened data with a rotation is still whitened. To simplify notation, we assume $\mathbf{Z}$ is centered by $\mathbf{Z}:=\mathbf{Z}(\mathbf{I}-\frac{1}{m}\mathbf{1}\mathbf{1}^\top)$. Ermolov et al. [16] propose W-MSE that uses Cholesky decomposition (CD) whitening: $\Phi_{CD}(\mathbf{Z})=\mathbf{L}^{-1}\mathbf{Z}$ in (4), where $\mathbf{L}$ is a lower triangular matrix from the Cholesky decomposition, with $\mathbf{L}\mathbf{L}^\top=\Sigma$. Here $\Sigma=\frac{1}{m}\mathbf{Z}\mathbf{Z}^\top$ is the covariance matrix of the embedding. Hua et al. [13] use zero-phase component analysis (ZCA) whitening [17] in (4): $\Phi_{ZCA}=\mathbf{U}\Lambda^{-\frac{1}{2}}\mathbf{U}^\top$, where $\Lambda=\text{diag}(\sigma_1,\ldots,\sigma_{d_z})$ and $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_{d_z}]$ are the eigenvalues and associated eigenvectors of Σ, i.e., $\mathbf{U}\Lambda\mathbf{U}^\top=\Sigma$. Another commonly used method is the principal components analysis (PCA) whitening: $\Phi_{PCA}=\Lambda^{-\frac{1}{2}}\mathbf{U}^\top$ [17], [18].

IV. Empirical Analysis of Whitening Loss

In this section, we empirically analyze the effects of different whitening transformations $\Phi(\cdot)$ used in (4) for SSL. In addition, we study the performance of different features (including encoding $\mathbf{H}$, embedding $\mathbf{Z}$, and the whitened output $\widehat{\mathbf{Z}}$) used as a representation for evaluation. For illustration, we first define the rank and stable rank [19] of a matrix:

Definition 4.1. Given a matrix $\mathbf{A}\in\mathbb{R}^{d\times m}$, $d\le m$, we denote $\{\lambda_1,\ldots,\lambda_d\}$ the singular values of $\mathbf{A}$ in a descent order with convention $\lambda_1>0$. The *rank* of $\mathbf{A}$ is the number of its non-zero singular values, denoted as $Rank(\mathbf{A})=\sum_{i=1}^d\mathbb{I}(\lambda_i>0)$, where $\mathbb{I}(\cdot)$ is the indicator function. The *stable rank* of $\mathbf{A}$ is denoted as $r(\mathbf{A})=\frac{\sum_{i=1}^d\lambda_i}{\lambda_1}$.

By definition, $Rank(\mathbf{A})$ can be a good indicator to evaluate the extent of dimensional collapse of $\mathbf{A}$, and $r(\mathbf{A})$ can be an indicator to evaluate the extent of whitening of $\mathbf{A}$. It can be shown that $r(\mathbf{A})\le Rank(\mathbf{A})\le d$ [19]. Note that if $\mathbf{A}$ is fully whitened with covariance matrix $\mathbf{A}\mathbf{A}^\top=m\mathbf{I}$, we have $r(\mathbf{A})=Rank(\mathbf{A})=d$. We also define normalized rank as $\widehat{Rank}(\mathbf{A})=\frac{Rank(\mathbf{A})}{d}$ and normalized stable rank as $\widehat{r}(\mathbf{A})=\frac{r(\mathbf{A})}{d}$, for comparing the extent of dimensional collapse and whitening of matrices with different dimensions, respectively.

A. PCA Whitening Fails to Avoid Dimensional Collapse

We evaluate the effects of ZCA, CD, and PCA transformations for whitening loss on CIFAR-10 using the standard setup for SSL (see Section VIII for details). In addition, we provide the result of batch normalization (BN) that only performs standardization

[1]The dual view formulation can be extended to s different views [16].

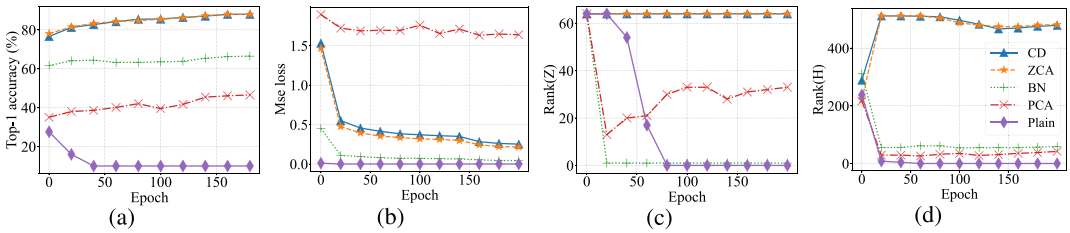


Fig. 2. Effects of different whitening transformations for SSL. We use the ResNet-18 as the encoder (the dimension of representation is 512.), a two-layer MLP with ReLU and BN appended as the projector (the dimension of embedding is 64). The model is trained on CIFAR-10 for 200 epochs with a batch size of 256 using Adam optimizer [44] and standard data argumentation. We show (a) the linear evaluation accuracy; (b) the training loss; (c) the rank of embedding; and (d) the rank of encoding.

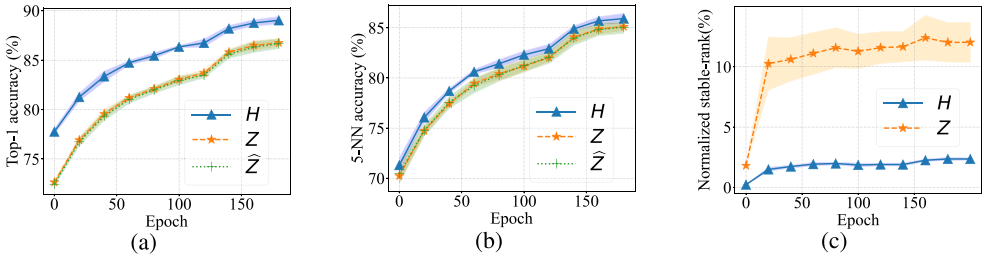


Fig. 3. Comparisons of features when using encoding $\mathbf{H}$, embedding $\mathbf{Z}$, and whitened output $\widehat{\mathbf{Z}}$ respectively. We use the same experimental setup as Fig. 2. We show (a) the linear evaluation accuracy; (b) the k-NN accuracy; and (c) the normalized stable rank for comparing the extent of whitening (note that the normalized stable rank of $\widehat{\mathbf{Z}}$ is always 100% during training, and we omit it for clarity). The results are averaged by five random seeds, with standard deviation shown using a shaded region.

without decorrelating the axes and the 'Plain' method that imposes the loss directly on embedding.

We use the ResNet-18 as the encoder (the dimension of encoding is 512), a two-layer MLP with ReLU and BN appended as the projector (the dimensions of the hidden layer and embedding are 1024 and 64). The model is trained on CIFAR-10 for 200 epochs with a batch size of 256, using Adam optimizer [44] with a learning rate of 3×10^{-3}, and learning rate warm-up for the first 500 iterations and a 0.2 learning rate drop at the last 50 and 25 epochs. The weight decay is set as 10^{-6}. All transformations are performed with two positives extracted per image with standard data argumentation (see Section VIII for details). We use the same evaluation protocol as in *W-MSE* [16].

Fig. 2 shows that naively training a Siamese network ('Plain') results in collapse both on the embedding (Fig. 2(c)) and encoding (Fig. 2(d)), which significantly affects the performance (Fig. 2(a)) although the training loss becomes close to zero (Fig. 2(b)). An extra BN imposed on the embedding prevents collapse to a point. However, it suffers from dimensional collapse where the rank of embedding and encoding is significantly low, which also negatively affects the performance. ZCA and CD whitening both maintain a high rank of embedding and encoding by decorrelating the axes, ensuring high linear evaluation accuracy. However, we note that PCA whitening shows significantly different behaviors as it cannot decrease the loss or avoid dimensional collapse. These results motivate us to analyze how whitening loss can be effectively used for SSL (see Sections V and VI).

B. *Whitened Output is Not a Good Representation*

As introduced above, the motivation of whitening loss for SSL is that the whitening operation can remove the correlation among axes [13] and a whitened representation ensures that the examples scattered in a spherical distribution [16], which is sufficient to avoid collapse. As such, one should use the whitened output $\widehat{\mathbf{Z}}$ as the representation for downstream tasks rather than the encoding $\mathbf{H}$ that is commonly used. This raises questions about whether $\mathbf{H}$ is well whitened and whether the whitened output is a good representation.

We conduct experiments to compare the performances of whitening loss when using $\mathbf{H}$, $\mathbf{Z}$ and $\widehat{\mathbf{Z}}$ as representations for evaluation, respectively. Fig. 3 shows that using whitened output $\widehat{\mathbf{Z}}$ as a representation has significantly worse performance than using $\mathbf{H}$. Furthermore, the normalized stable rank of $\mathbf{H}$ is significantly smaller than 100%, which suggests that $\mathbf{H}$ is not well whitened. These results show that the whitened output could not be a good representation.

V. Theoretical Analysis of Whitening Loss

In this section, we first decompose the whitening loss in the symmetric formulation into two asymmetric losses. Based on the decomposed formulation, we theoretically show that whitening loss indeed imposes a full-rank constraint on the embedding from the perspective of optimization and further theoretically demonstrate that the stable rank of the embedding is invariant during training by gradient descent, given the assumption that embedding is updated with an infinitely small learning rate.

A. Asymmetric Decomposition of Whitening Loss

For clarity, we use the mini-batch input with the size of m. Given one mini-batch input $\mathbf{X}$ with two augmented views, (4) can be formulated as:

$$\mathcal{L}(\mathbf{X}) = \frac{1}{m}\|\widehat{\mathbf{Z}}_1 - \widehat{\mathbf{Z}}_2\|_F^2. \tag{5}$$

Let us consider a proxy loss described as:

$$\mathcal{L}'(\mathbf{X}) = \underbrace{\frac{1}{m}\|\widehat{\mathbf{Z}}_1 - (\widehat{\mathbf{Z}}_2)_{st}\|_F^2}_{\mathcal{L}'_1} + \underbrace{\frac{1}{m}\|(\widehat{\mathbf{Z}}_1)_{st} - \widehat{\mathbf{Z}}_2\|_F^2}_{\mathcal{L}'_2}, \tag{6}$$

where $(\cdot)_{st}$ indicates the stop-gradient operation. It is easy to demonstrate that $\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}'}{\partial \theta}$ (see supplementary material for proof). That is, the optimization dynamics of $\mathcal{L}$ is equivalent to $\mathcal{L}'$. By looking into the first term of (6), we have:

$$\mathcal{L}'_1 = \frac{1}{m}\|\phi(\mathbf{Z}_1)\mathbf{Z}_1 - (\widehat{\mathbf{Z}}_2)_{st}\|_F^2. \tag{7}$$

Here, we see $\phi(\mathbf{Z}_1)$ as a predictor that depends on $\mathbf{Z}_1$ during forward propagation, and $\widehat{\mathbf{Z}}_2$ as a whitened target with $r(\widehat{\mathbf{Z}}_2) = Rank(\widehat{\mathbf{Z}}_2) = d_z$. In the following, our theoretical analysis is based on $\mathcal{L}'_1$, and similar analysis also applies to $\mathcal{L}'_2$.

B. Full-Rank Constraints on Embedding

Based on (7), minimizing $\mathcal{L}'_1$ only requires the embedding $\mathbf{Z}_1$ to be full-rank with $Rank(\widehat{\mathbf{Z}}_1) = d_z$, as stated by the following theorem.

Theorem 1: (Full-rank constraints on embedding). Let $\mathbb{A} = \arg\min_{\mathbf{Z}_1} \mathcal{L}'_1(\mathbf{Z}_1)$. We have that $\mathbb{A}$ is not an empty set, and $\forall \mathbf{Z}_1 \in \mathbb{A}$, $\mathbf{Z}_1$ is full-rank.

Proof: Since $\mathcal{L}'_1 \geq 0$, we have $\mathbb{A} = \{\mathbf{Z}_1 | \mathcal{L}'_1(\mathbf{Z}_1) = 0\}$. It is easy to validate that $\mathcal{L}'_1(\widehat{\mathbf{Z}}_2) = 0$, and we have $\widehat{\mathbf{Z}}_2 \in \mathbb{A}$. Therefore, $\mathbb{A}$ is not an empty set.

Next, we prove that $\forall \mathbf{Z}_1 \in \mathbb{A}$, $\mathbf{Z}_1$ is full-rank. We assume that for any $\mathbf{Z}_1 \in \mathbb{A}$ and $\mathbf{Z}_1$ is not a full-rank matrix, i.e., $Rank(\mathbf{Z}_1) < d_z$. We have $Rank(\phi(\mathbf{Z}_1)\mathbf{Z}_1) \leq Rank(\mathbf{Z}_1) < d_z$. We thus have that $\phi(\mathbf{Z}_1)\mathbf{Z}_1$ is not a full-rank matrix. As such, it is not possible for $\phi(\mathbf{Z}_1)\mathbf{Z}_1 = \widehat{\mathbf{Z}}_2$ since $\widehat{\mathbf{Z}}_2$ is a full-rank matrix. So $\mathcal{L}'_1(\mathbf{Z}_1) > 0$, which is contradictory to $\mathbf{Z}_1 \in \mathbb{A}$. Therefore, we have $\forall \mathbf{Z}_1 \in \mathbb{A}$, $\mathbf{Z}_1$ is full-rank. □

Theorem 1 states that minimizing $\mathcal{L}'_1$ only requires the embedding $\mathbf{Z}_1$ to be full-rank with $Rank(\widehat{\mathbf{Z}}_1) = d_z$, and does not necessarily impose the constraints on $\mathbf{Z}_1$ to be whitened with $r(\mathbf{Z}_1) = d_z$. Similar analysis also applies to $\mathcal{L}'_2$ and minimizing $\mathcal{L}'_2$ requires $\mathbf{Z}_2$ to be full-rank. Therefore, BW-based methods shown in (4) do not impose whitening constraints on the embedding as formulated in (3), but they only require the embedding to be *full-rank*. If we consider the objective $\mathcal{L}''_1 = \frac{1}{m}\|\mathbf{Z}_1 - (\widehat{\mathbf{Z}}_2)_{st}\|_F^2$ by removing the whitening transformation $\phi(\mathbf{Z}_1)$ in $\mathcal{L}'_1$, we find that minimizing $\mathcal{L}''_1$ requires the embedding $\mathbf{Z}_1$ to be whitened (i.e., encouraging $\mathbf{Z}_1$ to match the whitened target $\widehat{\mathbf{Z}}_2$). This constraint is similar to the whitening penalty $\|\frac{1}{m}\mathbf{Z}_1\mathbf{Z}_1^\top - \lambda\mathbf{I}\|_F^2$ in VICReg [15], as will be discussed in Section VI.

The full-rank constraint is also sufficient to avoid dimensional collapse for embedding, even though it is weaker than whitening constraint. Furthermore, recent works [40], [41], [42] demonstrate that a good representation should possess a high rank [42], and its distribution of singular values in the learned representation should follow a power-law distribution with an appropriate decay coefficient [40], [41]. These findings suggest that the full-rank constraint is superior to the whitening constraint. The full-rank constraint offers a more flexible solution space for singular value distributions, whereas the whitening constraint tends to encourage all singular values to converge to one.

Proposition 1: Let $\mathbb{A} = \arg\min_{\mathbf{Z}_1} \mathcal{L}'_1(\mathbf{Z}_1)$. For any $\{\lambda_i\}_{i=1}^{d_z}$ with $\lambda_1 \geq \lambda_2 \geq, \ldots, \lambda_{d_z} > 0$, we construct $\widetilde{\mathbb{A}} = \{\mathbf{Z}_1 | \mathbf{Z}_1 = \mathbf{U}_2\ \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d_z})\ \mathbf{V}_2^\top$, where $\mathbf{U}_2 \in \mathbb{R}^{d_z \times d_z}$ and $\mathbf{V}_2 \in \mathbb{R}^{m \times d_z}$ are from the SVD of $\widehat{\mathbf{Z}}_2$, i.e., $\mathbf{U}_2(\sqrt{m}\mathbf{I})\mathbf{V}_2^\top = \widehat{\mathbf{Z}}_2$. When we use ZCA whitening, we have $\widetilde{\mathbb{A}} \subseteq \mathbb{A}$.

Proof: For any $\{\lambda_i\}_{i=1}^{d_z}$ with $\lambda_1 \geq \lambda_2 \geq, \ldots, \lambda_{d_z} > 0$, let $\mathbf{Z}_1 = \mathbf{U}_2\ \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d_z})\ \mathbf{V}_2^\top$, we now prove that $\phi(\mathbf{Z}_1)\mathbf{Z}_1 = \widehat{\mathbf{Z}}_2$ when using ZCA whitening. We know $\phi(\mathbf{Z}_1) = \Phi_{ZCA} = \mathbf{U}\Lambda^{-\frac{1}{2}}\mathbf{U}^\top$, where $\Lambda = \mathrm{diag}(\sigma_1, \ldots, \sigma_{d_z})$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_z}]$ are the eigenvalues and associated eigenvectors of the covariance matrix Σ of $\mathbf{Z}_1$. We know that $\Sigma = \frac{1}{m}\mathbf{Z}_1\mathbf{Z}_1^\top = \mathbf{U}_2\ \mathrm{diag}(\lambda_1^2/m, \lambda_2^2/m, \ldots, \lambda_{d_z}^2/m)\ \mathbf{U}_2^\top$. Since the eigen decomposition of Σ is unique, we have $\phi(\mathbf{Z}_1) = \mathbf{U}_2\ \mathrm{diag}(\sqrt{m}/\lambda_1, \sqrt{m}/\lambda_2, \ldots, \sqrt{m}/\lambda_{d_z})\ \mathbf{U}_2^\top$. Therefore, $\phi(\mathbf{Z}_1)\mathbf{Z}_1 = \mathbf{U}_2\ \mathrm{diag}(\sqrt{m}/\lambda_1, \sqrt{m}/\lambda_2, \ldots, \sqrt{m}/\lambda_{d_z})\ \mathbf{U}_2^\top\mathbf{U}_2\ \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d_z})\mathbf{V}_2^\top = \mathbf{U}_2(\sqrt{m}\mathbf{I})\mathbf{V}_2^\top = \widehat{\mathbf{Z}}_2$. We thus have $\widetilde{\mathbb{A}} \subseteq \mathbb{A}$. □

Proposition 1 constructs the potential solution space when using ZCA whitening for whitening loss. From this proposition, it becomes evident that minimizing $\mathcal{L}'_1$ requires the left-singular vectors and right-singular vectors of $\mathbf{Z}_1$ to match the ones of the whitened target $\widehat{\mathbf{Z}}_2$, but leaves the (non-zero) singular values of $\mathbf{Z}_1$ to be free. In the subsequent section, we will delve into the evolution of the singular values of the embedding during training.

C. Training Dynamics of Embedding

We analyze the training dynamics of embedding and show that the gradient of the whitening loss $\mathcal{L}$ w.r.t. the embedding $\boldsymbol{Z}_1$ is orthogonal to the gradient of any singular value λ of the embedding w.r.t. the embedding itself.

Theorem 2: $\left(\textit{Orthogonality between } \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1} \textit{ and } \frac{\partial \lambda}{\partial \boldsymbol{Z}_1}\right)$. Denote the inner product of a $d \times m$ matrix $\boldsymbol{A} = \{a_{ij}\}_{d \times m}$ and $\boldsymbol{B} = \{b_{ij}\}_{d \times m}$ as $< \boldsymbol{A}, \boldsymbol{B} >= \sum_{i,j} a_{ij}b_{ij}$. For any singular value λ of the embedding matrix $\boldsymbol{Z}_1$, we have

$$< \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}, \frac{\partial \lambda}{\partial \boldsymbol{Z}_1} >= 0. \tag{8}$$

We leave the detailed derivations of Theorem 2 in the supplementary material. Theorem 2 shows that the vector $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ in the embedding space ($\boldsymbol{Z}_1$) is perpendicular to the vector $\frac{\partial \lambda}{\partial \boldsymbol{Z}_1}$ from the geometric perspective. Considering the landscape of any singular value λ w.r.t. the embedding $\boldsymbol{Z}_1$, if we update $\boldsymbol{Z}_1$

along the direction of $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$, the singular value is likely to remain invariant intuitively, since $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ is parallel to the contour[2] of λ

Based on Theorem 2, it is easy to derive that the gradient of the stable rank of embedding $\boldsymbol{Z}_1$ w.r.t. the embedding itself is orthogonal to that of the whitening loss $\mathcal{L}$ w.r.t. the embedding.

Theorem 3: $\left(\textit{Orthogonality between} \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1} \textit{ and } \frac{\partial r(\boldsymbol{Z}_1)}{\partial \boldsymbol{Z}_1}\right)$. Let $\lambda_1 \geq \cdots \geq \lambda_{d_z}$ be the singular values of $\boldsymbol{Z}_1$, and the stable rank of $\boldsymbol{Z}_1$ be $r(\boldsymbol{Z}_1) = \frac{\lambda_1 + \cdots + \lambda_{d_z}}{\lambda_1}$. We have

$$< \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}, \frac{\partial r(\boldsymbol{Z}_1)}{\partial \boldsymbol{Z}_1} >= 0. \tag{9}$$

Proof: Based on the definition of stable rank, we have

$$\frac{\partial r(\boldsymbol{Z}_1)}{\partial \boldsymbol{\lambda}} = \frac{1}{\lambda_1^2}\left[-\sum_{i=2}^{d_z}\lambda_i, \lambda_1, \ldots, \lambda_1\right]^\top. \tag{10}$$

According to (8), we have

$$\begin{aligned} < \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}, \frac{\partial r(\boldsymbol{Z}_1)}{\partial \boldsymbol{Z}_1} > &= \sum_{i=1}^{d_z} < \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}, \frac{\partial r(\boldsymbol{Z}_1)}{\partial \lambda_i}\frac{\partial \lambda_i}{\partial \boldsymbol{Z}_1} > \\ &= \sum_{i=1}^{d_z} \frac{\partial r(\boldsymbol{Z}_1)}{\partial \lambda_i} < \frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}, \frac{\partial \lambda_i}{\partial \boldsymbol{Z}_1} > \\ &= 0. \end{aligned} \tag{11}$$

□

Theorem 3 shows that the vector $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ in the embedding space $(\boldsymbol{Z}_1)$ is perpendicular to the vector $\frac{\partial r(\boldsymbol{Z}_1)}{\partial \boldsymbol{Z}_1}$ from the geometric perspective. Considering the landscape of stable rank $r(\boldsymbol{Z}_1)$ w.r.t. the embedding $\boldsymbol{Z}_1$, if we update $\boldsymbol{Z}_1$ along the direction of $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$, the stable rank is likely to remain invariant intuitively, since $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ is parallel to the contour of $r(\boldsymbol{Z}_1)$. We further clarify this intuition with rigorous evidences by the following theorem.

Theorem 4: (Invariance property of $r(\boldsymbol{Z}_1)$ during training, under certain assumptions). We let $\mathcal{L}^{(0)}$ and $\mathcal{L}^{(T)}$ be the initial and final loss, where T is the iteration number. The loss is updated using the learning rate η. Given a constant $\Delta\mathcal{L} = \mathcal{L}^{(T)} - \mathcal{L}^{(0)}$, the stable rank is invariant during the course of training if we use gradient descent under the assumptions that: 1) the learning rate $\eta \to 0$; 2) the L-2 norm of $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ and the operator norm of $\frac{\partial^2 r(\boldsymbol{Z}_1)}{\partial \boldsymbol{Z}_1^2}$ have an upper bound C and a upper bound M, respectively.

Proof: To simplify the notation, we denote $\boldsymbol{Z}_1$ as a vector $\boldsymbol{z}$, and denote $R = r(\boldsymbol{Z}_1)$. Let $\boldsymbol{z}^{(0)}$ be the initial state, and $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(T)}$ be the (arbitrary) T iteration states of $\boldsymbol{z}$ during training. As $\boldsymbol{z}$ changes, we have the loss sequences $\mathcal{L}^{(0)}, \mathcal{L}^{(1)}, \ldots, \mathcal{L}^{(T)}$ and the stable rank sequences $R^{(0)}, R^{(1)}, \ldots, R^{(T)}$. Given a state $\boldsymbol{z}^{(t)}$, a gradient descent method seeks to update the state along the negative direction of gradient $\boldsymbol{g}^{(t)} = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}$. We thus have $\Delta\boldsymbol{z}^{(t)} = \boldsymbol{z}^{(t+1)} - \boldsymbol{z}^{(t)} = \eta \boldsymbol{g}^{(t)}$. Denote $d = \max_t \|\Delta \boldsymbol{z}^{(t)}\|_2$. Since $\|\Delta \boldsymbol{z}^{(t)}\|_2 = \eta \|\boldsymbol{g}^{(t)}\|_2 \leq C\eta$, we have $d \to 0$ when $\eta \to 0$. Since $\mathcal{L}$ is differentiable w.r.t. $\boldsymbol{z}$, according to the definition of the second type of curve integral, we have

$$\begin{aligned} \mathcal{L}^{(T)} - \mathcal{L}^{(0)} &= \int_{\boldsymbol{z}^{(0)}}^{\boldsymbol{z}^{(T)}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}} \cdot d\boldsymbol{z} \\ &= \lim_{d \to 0} \sum_{t=0}^{T-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}} \cdot \Delta \boldsymbol{z}^{(t)} \\ &= \lim_{\eta \to 0} \sum_{t=0}^{T-1} -\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right) \eta. \end{aligned} \tag{12}$$

On the other hand, we have

$$\begin{aligned} &|R^{(T)} - R^{(0)}| \\ &\leq \sum_{t=0}^{T-1} \left| R^{(t+1)} - R^{(t)} \right| \\ &= \sum_{t=0}^{T-1} \left| \left(\frac{\partial R}{\partial \boldsymbol{z}^{(t)}}\right)^\top (\Delta \boldsymbol{z}^{(t)}) + (\Delta \boldsymbol{z}^{(t)})^\top \left(\frac{\partial^2 R}{\partial [\bar{\boldsymbol{z}}^{(t)}]^2}\right) (\Delta \boldsymbol{z}_t) \right|, \end{aligned} \tag{13}$$

where $\bar{\boldsymbol{z}}^{(t)} = \mu^{(t)} \boldsymbol{z}^{(t)} + (1 - \mu^{(t)}) \boldsymbol{z}^{(t+1)}, \mu^{(t)} \in [0, 1]$, according to Taylor's mean value theorem. Since $\Delta \boldsymbol{z}^{(t)} = -\eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}$ and $\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial R}{\partial \boldsymbol{z}^{(t)}}\right) = 0$ (according to Theorem 3), we have

$$\begin{aligned} &\sum_{t=0}^{T-1} \left| \left(\frac{\partial R}{\partial \boldsymbol{z}^{(t)}}\right)^\top (\Delta \boldsymbol{z}^{(t)}) + (\Delta \boldsymbol{z}^{(t)})^\top \left(\frac{\partial^2 R}{\partial [\bar{\boldsymbol{z}}^{(t)}]^2}\right) (\Delta \boldsymbol{z}^{(t)}) \right| \\ &= \sum_{t=0}^{T-1} \left| \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial^2 R}{\partial [\bar{\boldsymbol{z}}^{(t)}]^2}\right) \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right) \eta^2 \right|. \end{aligned} \tag{14}$$

Here, we show $((\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}})^\top \frac{\partial^2 R}{\partial \bar{\boldsymbol{z}}^2} \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}})$ can be up-bounded by the following Lemma 5.1.

Lemma 5.1:

$$\left| \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}\right)^\top \left(\frac{\partial^2 R}{\partial \bar{\boldsymbol{z}}^2}\right) \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}\right) \right| \leq M \left| \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}\right)^\top \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}\right) \right|. \tag{15}$$

Proof: Consider the extreme value of $f(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$, where $\boldsymbol{A}$ is a symmetric matrix. We denote that the value of $f(\boldsymbol{x})$ is invariable when adding the constraint $\boldsymbol{x}^\top \boldsymbol{x} = 1$. We consider the Lagrange function of this constrained optimization problem

$$h(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} - \lambda(\boldsymbol{x}^\top \boldsymbol{x} - 1), \tag{16}$$

whose extremum conditions are

$$\begin{cases} \boldsymbol{A}\boldsymbol{x} - \lambda \boldsymbol{x} = 0, \\ \boldsymbol{x}^\top \boldsymbol{x} = 1. \end{cases} \tag{17}$$

Consequently, we have $\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} = \lambda \boldsymbol{x}^\top \boldsymbol{x} = \lambda$. Since λ is an eigenvalue of $\boldsymbol{A}$, the extreme value of $f(\boldsymbol{x})$ must be an

[2]This is because $\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_1}$ is perpendicular to $\frac{\partial \lambda}{\partial \boldsymbol{Z}_1}$.

eigenvalue of $\boldsymbol{A}$. Thus $\lambda_{min}(\boldsymbol{A}) \leq f(\boldsymbol{x}) \leq \lambda_{max}(\boldsymbol{A})$. Since $|\lambda(\boldsymbol{A})| \leq \|\boldsymbol{A}\|$, we have $|f(\boldsymbol{x})| \leq \|\boldsymbol{A}\|$. Let $\boldsymbol{x} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}, \boldsymbol{A} = \frac{\partial^2 R}{\partial \bar{\boldsymbol{z}}^2}$. Since $\|\frac{\partial^2 R}{\partial \bar{\boldsymbol{z}}^2}\| \leq M$, we thus prove Lemma 5.1. □

According to Lemma 5.1, we have

$$\begin{aligned} |R^{(T)} - R^{(0)}| &\leq \sum_{t=0}^{T-1} \left| \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial^2 R}{\partial [\bar{\boldsymbol{z}}^{(t)}]^2}\right) \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right) \eta^2 \right| \\ &\leq M\eta \sum_{t=0}^{T-1} \left| \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right) \eta \right| \\ &= M\eta \sum_{t=0}^{T-1} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right)^\top \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^{(t)}}\right) \eta \\ &= M\eta(\mathcal{L}^{(0)} - \mathcal{L}^{(T)}). \end{aligned} \tag{18}$$

Since M and $\mathcal{L}^{(0)} - \mathcal{L}^{(T)}$ are two positive constants, and let $\eta \to 0$, we have $|R^{(T)} - R^{(0)}| \to 0$. Note that T is arbitrary in our derivation. It shows the stable rank R is invariant during training when using gradient descent. □

Note that assumption 2 of Theorem 4 is satisfied for a loss function $\mathcal{L}$ with the bounded Lipschitz constant, and the embedding $\boldsymbol{Z}_1$ is full-rank. Therefore, assumption 2 is usually satisfied for whitening loss during training. In practice, the embedding is full-rank in the initial state, likely caused by the weights being initialized with Gaussian distribution. We also observe that the stable rank is almost invariant in our toy experiments, given a small learning rate to update the embedding using gradient descent.

VI. Discussions

Our analysis suggests that whitening loss in its symmetric formulation ((5)) can be decomposed into two asymmetric losses ((6)), where each asymmetric loss requires an online network to match a whitened target. This mechanism provides a pivot connection to other methods and a clue as to why PCA whitening fails to avoid dimensional collapse for SSL.

Connection to Asymmetric Methods: The asymmetric formulation of whitening loss shown in (7) bears resemblance to those asymmetry methods without negative pairs, e.g., SimSiam [5]. In these methods, an extra predictor is incorporated, and the stop-gradient is essential for avoid collapse. Specifically, SimSiam uses the objective as:

$$\begin{aligned} \mathcal{L}(\mathbf{X}) = \frac{1}{m}\|P_{\theta_p}(\cdot) \circ \mathbf{Z}_1 - (\mathbf{Z}_2)_{st}\|_F^2 + \frac{1}{m}\|P_{\theta_p}(\cdot) \circ \mathbf{Z}_2 \\ - (\mathbf{Z}_1)_{st}\|_F^2, \end{aligned} \tag{19}$$

where $P_{\theta_p}(\cdot)$ is the predictor with learnable parameters θ_p. By contrasting (7) and the first term of (19), we find that: 1) BW-based whitening loss ensures a whitened target $\widehat{\mathbf{Z}}_2$, while SimSiam does not put a constraint on the target $\mathbf{Z}_2$; 2) SimSiam uses a learnable predictor $P_{\theta_p}(\cdot)$, which is shown to empirically avoid collapse by matching the rank of the covariance matrix by back-propagation [33], while BW-based whitening loss has an implicit predictor $\phi(\mathbf{Z}_1)$ depending on the input itself, a full-rank matrix by design. As such, BW-based whitening loss can surely avoid collapse if the loss converges well, while SimSiam can not provide such a guarantee to avoid collapse. Similar analysis also applies to BYOL [4], except that BYOL uses a momentum target network for providing a target signal.

Connection to Soft Whitening: VICReg [15] also encourages whitened embedding produced from different views, but by imposing a whitening penalty as a regularization on the embedding, which is called soft whitening. Given a mini-batch input, the objective of VICReg is as follows:[3]

$$\mathcal{L}(\mathbf{X}) = \frac{1}{m}\|\mathbf{Z}_1 - \mathbf{Z}_2\|_F^2 + \alpha \sum_{i=1}^{2} \left(\left\| \frac{1}{m}\mathbf{Z}_i\mathbf{Z}_i^\top - \lambda \mathbf{I} \right\|_F^2 \right), \tag{20}$$

where $\alpha \geq 0$ is the penalty factor. Similarly, we can use a proxy loss for VICReg, and considering its term corresponding to optimizing $\mathbf{Z}_1$ only (similar to (7)), we have:

$$\mathcal{L}'_{VICReg}(\mathbf{X}) = \frac{1}{m}\|\mathbf{Z}_1 - (\mathbf{Z}_2)_{st}\|_F^2 + \alpha \|\frac{1}{m}\mathbf{Z}_1\mathbf{Z}_1^\top - \lambda \mathbf{I}\|_F^2. \tag{21}$$

Based on this formulation, we observe that VICReg requires embedding $\mathbf{Z}_1$ to be whitened by, 1) the additional whitening penalty, and 2) fitting the (expected) whitened targets $\mathbf{Z}_2$. By contrasting (7) and (21), we highlight that the so-called hard whitening methods, e.g., W-MSE [16], only impose full-rank constraints on the embedding, while soft whitening methods indeed impose whitening constraints. Similar analysis also applies to Barlow Twins [36], except that the whitening/decorrelation penalty is imposed on the cross-covariance matrix of embedding from different views.

Connection to Other Non-contrastive Methods: SwAV [29] uses a swapped prediction mechanism where the cluster assignment (code) of a view is predicted from the representation of another view by minimizing the following objective:

$$\mathcal{L}(\mathbf{X}) = \ell(\mathbf{C}^\top \mathbf{Z}_1, (\mathbf{Q}_2)_{st}) + \ell(\mathbf{C}^\top \mathbf{Z}_2, (\mathbf{Q}_1)_{st}). \tag{22}$$

Here, $\mathbf{C}$ is the prototype matrix learned by back-propagation, $\mathbf{Q}_i$ is the predicted code with equal-partition and high-entropy constraints, and SwAV uses cross-entropy loss as $\ell(\cdot, \cdot)$ to match the distributions. The constraints on $\mathbf{Q}_i$ are approximately satisfied during optimization by using the iterative Sinkhorn-Knopp algorithm conditioned on the input $\mathbf{C}^\top \mathbf{Z}_i$. SwAV explicitly uses stop-gradient when it calculates the target $\mathbf{Q}_i$. By contrasting (7) and the first term of (22), we find that: 1) SwAV can be viewed as an online network to match a target with constraints, like BW-based whitening loss, even though the constraints imposed on the targets between them are different; 2) From the perspective of asymmetric structure, SwAV indeed uses a linear predictor $\mathbf{C}^\top$ that is also learned by back-propagation like SimSiam, while BW-based whitening loss has an implicit predictor $\phi(\mathbf{Z}_1)$ depending on the input itself. Similar analysis also applies to DINO [31], which further simplifies the formulation of SwAV by removing the prototype matrix and directly matching the output of another view from the view of knowledge distillation.

[3]Note the slight difference where VICReg uses margin loss on the diagonal of covariance, while our notation uses MSE loss.

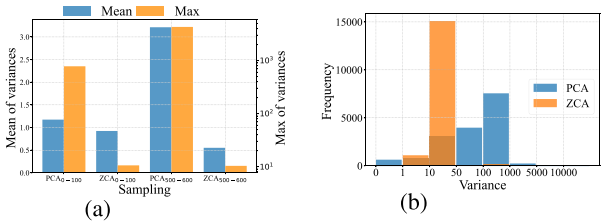


Fig. 4. Illustration of PCA-based whitening loss suffering from training instability. We use the same experimental setup as Fig. 2. Given a certain mini-batch input ($m = 2048$), we monitor its whitened output $\widehat{\mathbf{Z}}^t$ and whitening matrix Φ^t for each epoch t. We calculate the variance along the training epochs for each element of $\widehat{\mathbf{Z}}$ and Φ. We show (a) the mean and maximum of variances of $\widehat{\mathbf{Z}}$, noting that PCA_{0-100} indicates the variance of PCA whitened output is calculated along the first 100 epochs and (b) the histogram of variance of Φ.

DINO uses centering and sharpening operations to impose constraints on the target (output of another view). One significant difference between DINO and whitening loss is that DINO uses population statistics of centering calculated by moving average while whitening loss uses the mini-batch statistics of whitening.

Why PCA Whitening Fails to Avoid Dimensional Collapse? Based on (7), whitening loss can favorably enforce full-rank constraints on the embedding when the online network can match the whitened targets well. We experimentally show that PCA-based whitening loss provides a varying sequence of whitened targets during training, as shown in Fig. 4(a). It is difficult for the online network to match such a target signal with significant variation, resulting in a minimal decrease in the whitening loss (see Fig. 2). Furthermore, we observe that PCA-based whitening loss also has significantly varying whitening matrix sequences $\{\phi^t(\cdot)\}$ (Fig. 4(b)), even given the same input data. The findings agree with the observation in [4], [5], where an unstable predictor results in significant degenerate performance. Our observations also agree with the results in [17], [45] that PCA-based BW shows significantly large stochasticity. We note that ZCA whitening can provide relatively stable sequences of whitened targets and whitening matrix during training (Fig. 4), ensuring stable SSL training. This is likely due to the property of ZCA-based whitening that minimizes the total squared distance between the original and whitened variables [17], [18].

Why is Whitened Output not a Good Representation? A whitened output removes the correlation among axes [13] and ensures the examples are scattered in a spherical distribution [16], which bears some resemblance to contrastive learning, where different examples are pulled away. We conduct experiments to compare SimCLR [3], BYOL [4], VICReg [15], and W-MSE [16], and monitor the cosine similarity for all negative pairs, stable rank, and rank during training. Fig. 5 shows that all methods can achieve a high rank on the encoding. This is driven by the improved extent of whitening on the embedding. Furthermore, we observe that the negative cosine similarity decreases during the training while the extent of stable rank increases for all methods. This observation suggests that a representation with a stronger extent of whitening is more likely to have less similarity among different examples. We further conduct experiments to validate this, using VICReg with varying penalty factor α ((21)) to adjust the extent of whitening on embedding (Fig. 5(d)). Therefore, a whitened output leads to the state that all examples have dissimilar features. This state can break the potential manifold the examples in the same class belong to, which makes learning more difficult [41], [46]. Similar analysis for contrastive learning is also shown in [3], where classes represented by the projected output (embedding) are not well separated, compared to encoding.

VII. Channel Whitening With Random Group Partition

One main weakness of BW-based whitening loss is that the whitening operation requires the number of examples (mini-batch size) m to be larger than the size of channels d, to avoid numerical instability[4] [47], [48]. This requirement limits its usage in scenarios where a large batch of training data cannot be fit into the memory. Based on above analysis, the whitening loss can be viewed as an online learner matching a whitened target with all singular values being one. We note the key to whitening loss is that it conducts a transformation $\phi : \mathbf{Z} \rightarrow \widehat{\mathbf{Z}}$, ensuring that the singular values of $\widehat{\mathbf{Z}}$ are one. We thus propose channel whitening (CW) that ensures the examples in a mini-batch are orthogonal:

$$\begin{aligned} Centering : \mathbf{Z}_c &= \left(\mathbf{I} - \frac{1}{d}\mathbf{1}\cdot\mathbf{1}^\top\right)\mathbf{Z}, \\ Whitening : \widehat{\mathbf{Z}} &= \mathbf{Z}_c\Phi, \end{aligned} \tag{23}$$

where $\Phi \in \mathbb{R}^{m\times m}$ is the whitening matrix that is derived from the corresponding covariance matrix: $\Sigma' = \frac{1}{d-1}\mathbf{Z}_c^\top\mathbf{Z}_c$. In our implementation, we use ZCA whitening to obtain Φ. CW ensures the examples in a mini-batch are orthogonal to each other, with $\widehat{\mathbf{Z}}^\top\widehat{\mathbf{Z}} = \frac{1}{d-1}\mathbf{I}$. This means CW has the same ability as BW for SSL in avoiding the dimensional collapse by providing target $\widehat{\mathbf{Z}}$ whose singular values are one. More importantly, one significant advantage of CW is that it can obtain numerical stability when the batch size is small since the condition that $d > m$ can be obtained by design (e.g., we can set the channel number of embedding d to be larger than the batch size m). In addition, we find that CW can amplify the full-rank constraints on the embedding by dividing the channels/neurons into random groups, as we will illustrate.

Random Group Partition: Given the embedding $\mathbf{Z} \in \mathbb{R}^{d\times m}, d > m$, we divide it into $g \geq 1$ groups $\{\mathbf{Z}^{(i)} \in \mathbb{R}^{\frac{d}{g}\times m}\}_{i=1}^{g}$, where we assume that d is divisible by g and ensure $\frac{d}{g} > m$. We then perform CW on each $\mathbf{Z}^{(i)}, i = 1, \ldots, g$. Note that the ranks of $\mathbf{Z}$ and $\mathbf{Z}^{(i)}$ are all at most m. Therefore, CW with group partition provides g constraints with $Rank(\mathbf{Z}^{(i)}) = m$ on embedding, as opposed to CW without group partition that provides only one constraint with $Rank(\mathbf{Z}) = m$. Although CW with group partition can provide more full-rank constraints for mini-batch data, it can also make the population data correlated if the group partition is all the same during training, which decreases the rank and does not improve the performance in

[4] An empirical setting is $m = 2\,d$ that can obtain good performance as shown in [13], [16].

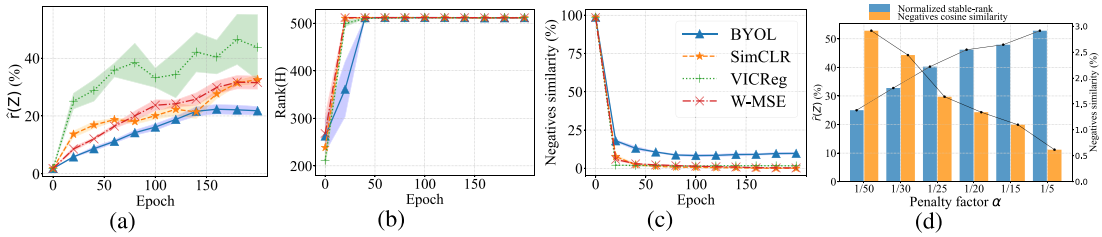


Fig. 5. Comparison of different SSL methods. We use the same experimental setup as Fig. 2. We show (a) the normalized stable rank of embedding; (b) the rank of encoding; (c) the negatives cosine similarity, calculated on the embeddings from all negative pairs (different examples). We also train VICReg [15] with varying penalty factor α to show the relationship between the normalized stable rank and negatives cosine similarity in (d). Here, we use the embedding dimension of 64. We have similar observations when using the embedding dimension of other numbers (e.g., 128 and 256).

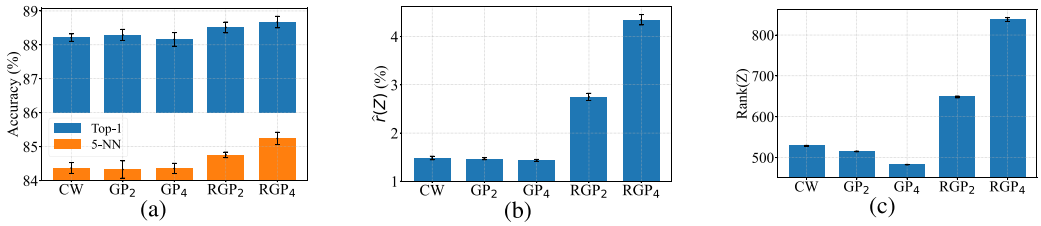


Fig. 6. Illustration of CW with random group partition. We use the same experimental setup as Fig. 2, except that we set the dimension of embedding as 2048 tailored for CW. We use 'GP$_2$' ('RGP$_2$') to indicate CW using group partition (random group partition), with a group number of 2. (a) The linear and k-NN accuracy; (b) The normalized stable rank of embedding; (c) The rank of embedding. All experiments are repeated five times, with standard deviation shown as error bars.

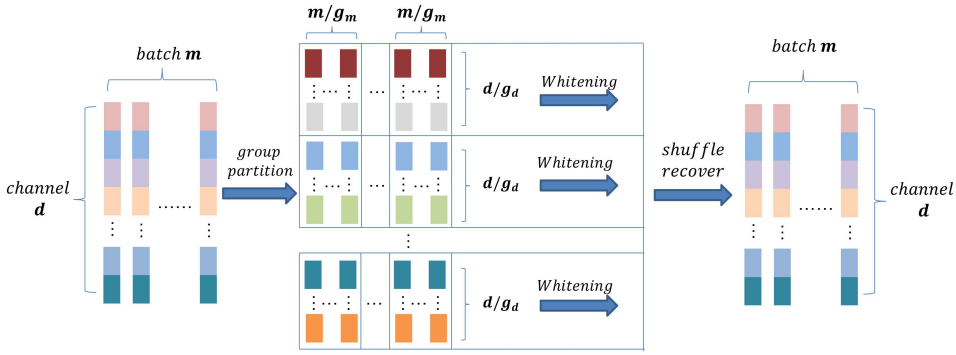


Fig. 7. General framework in group partition. We perform the group partition not only along the channel dimension but also the batch dimension. Given the embedding of mini-batch data, we randomly shuffle the data and divide the shuffled data into $g_m \cdot g_d$ groups. We then perform CW on each group and shuffle the data back accordingly.

accuracy by our experiments (Fig. 6). We find random group partition, which randomly divides the channels/neurons into groups for each iteration (mini-batch data), can alleviate this issue and achieve better performance, as shown in Fig. 6. We call our method as channel whitening with random group partition (CW-RGP), and provide the full algorithm and PyTorch-style code in supplementary material.

We note that Hua et al. [13] use a similar idea for BW, called Shuffled-DBN. It also divides the channels into groups randomly for each iteration and performs BW on each group $\mathbf{Z}^{(i)}, i = 1, \ldots, g$. However, Shuffled-DBN cannot well amplify the full-rank constraints by using more groups since BW-based methods require $m > \frac{d}{g}$ to avoid numerical instability. We further show that CW-RGP performs better than Shuffled-DBN in the subsequent experiments. These results can be attributed to the ability of CW-RGP to amplify the full-rank constraints by using groups.

General Framework in Group Partition: Based on the analysis in Section V-A that whitening loss can be viewed as a full-ranking constraint over the embedding of mini-batch data, we propose a general framework to divide the group for CW method. In this framework, we show that the group partition can be performed not only along the channel dimension but also the batch dimension. Formally, given the embedding of mini-batch data $\mathbf{Z} \in \mathbb{R}^{d \times m}, d > m$, we divide it into $g_m \cdot g_d$ groups $\{\mathbf{Z}^{(i,j)} \in \mathbb{R}^{\frac{d}{g_d} \times \frac{m}{g_m}} | i = 1, 2, \ldots, g_m, j = 1, 2, \ldots, g_d\}$(Fig. 7),

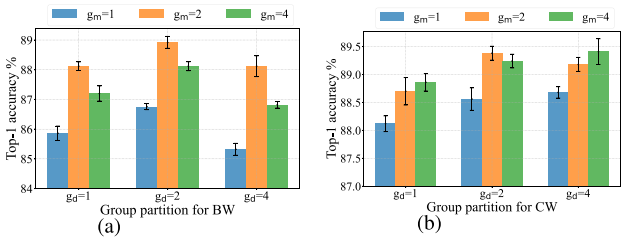


Fig. 8. Performance improvement of BW and CW, when searching a good group partition (g_m, g_d). (a) Results of BW, where we use embedding dimension $d = 64$ and batch size $m = 512$; (b) Results of BW, where we use embedding dimension $d = 2048$ and batch size $m = 256$. We can see BW with group partition (2,2) obtains the best performance, while CW with group partition (4,4) performs best.

where we assume that d (or m) is divisible by g_d (or g_m), and ensure $\frac{d}{g_d} > \frac{m}{g_m}$. We then perform CW on each $\mathbf{Z}^{(i,j)}$. Therefore, this general group partition provides $g_m \cdot g_d$ constraints with $Rank(\mathbf{Z}^{(i,j)}) = \frac{m}{g_m}$ on embedding. We also propose using random group partition along the channel and batch dimensions in this general framework. Note that CW-RGP is an instance of this framework if $g_m = 1$ with random group partition along the channel dimension.

Similarly, this general framework in group partition can also apply to BW methods [13], [16], except that we need to ensure $\frac{m}{g_m} > \frac{d}{g_d}$ for BW. We demonstrate that W-MSE with batch slicing [16] is an instance of this framework for BW if we use $g_d = 1$ with random group partition along the batch dimension. Furthermore, Shuffled-DBN [13] is an instance of this framework for BW if $g_m = 1$ with random group partition along channel dimension. We also show that we can improve the performance of BW (W-MSE/Shuffled-DBN) and CW (CW-RGP) methods when searching a good group partition (g_m, g_d) based on this framework in Fig. 8.

VIII. Experiments on Standard SSL Benchmark

In this section, we conduct experiments to validate the effectiveness of CW-RGP and state-of-the-art methods on the CIFAR-10, CIFAR-100 [49], STL-10 [50], TinyImageNet [51] and ImageNet [52] datasets. We also evaluate the effectiveness in transfer learning for a pre-trained model using CW-RGP. All experiments are carried out on a machine with 4 GPUs.

A. Evaluation for Classification

1) Evaluation on Small and Medium Size Datasets: We first conduct experiments on small and medium size datasets (including CIFAR-10, CIFAR-100, STL-10 and TinyImageNet), using the same experimental setups as W-MSE [16].

Encoder and Projector: We use the ResNet-18 [53] as the encoder, and the encoding dimension is 512. In addition, we use a 2-layers MLP as the projector: one hidden layer with BN and Relu applied to it and a linear layer as output. For the experiments on the CIFAR-10, CIFAR-100 and STL-10 datasets, the dimensions of the hidden layer in the projector and embedding are 1024 and 512. In the experiments of Tiny-ImageNet, the dimensions of the hidden layer of the projector and embedding are 2048 and 1024.

Image Transformation Details: Following the setups in [3], we transform images by extracting crops with a random size from 0.2 to 1.0 of the original area and an arbitrary aspect ratio from 3/4 to 4/3 of the original aspect ratio. The horizontal mirroring is applied with a probability of 0.5, and the color jittering configuration is (0.4, 0.4, 0.4, 0.1) with a probability of 0.8 and grayscaling with a probability of 0.1. For ImageNet-100, the crop size is from 0.08 to 1.0, jittering is strengthened to (0.8, 0.8, 0.8, 0.2), the grayscaling probability is 0.2, and Gaussian blurring is with a probability of 0.5. We use only one crop at testing time in all the experiments (standard protocol).

Optimizer and Learning Rate Schedule: We use the Adam optimizer [44]. In addition, we apply the same number of epochs and learning rate schedules to all the compared methods. For CIFAR-10 and CIFAR-100, we use 1,000 epochs with a learning rate of 3×10^{-3}; for STL-10, 2,000 epochs with a learning rate of 2×10^{-3}; for Tiny-ImageNet, 1000 epochs with a learning rate of 2×10^{-3}. We use a 0.2 learning rate drop at the last 50 and 25 epochs in these experiments, and the weight decay is 10^{-6}. In all experiments, we use learning rate warm-up for the first 500 iterations of the optimizer. We use a batch size of 512 for CW-RGP in the CIFAR-100, STL-10, and Tiny ImageNet experiments and 256 for the others.

Evaluation Protocol: We use the same evaluation setups as in *W-MSE* [16]: Training the linear classifier for 500 epochs using the Adam optimizer and labeled training set of each specific dataset, without data augmentation; the learning rate is exponentially decayed from 10^{-2} to 10^{-6}, and the weight decay is 5×10^{-6}. In addition, we evaluate the accuracy of a k-nearest neighbors classifier (k-NN, k = 5) in these experiments.

Our CW-RGP has the same advantages as W-MSE in exploiting different views. CW-RGP 2 and CW-RGP 4 indicate our methods with $s = 2$ and $s = 4$ positive views extracted per image, similar to *W-MSE* [16]. Some results of baselines in Table I are from [16], and others are from our implementations using the same training and evaluation settings as in [16] (some different hyper-parameter settings are shown in supplementary material).

CW-RGP obtains the highest accuracy on almost all the datasets except Tiny-ImageNet. In addition, CW-RGP with 4 views is generally better than 2, similar to W-MSE. These results show that CW-RGP is an effective SSL method. In addition, CW with random group partition performs better than BW methods (with random group partition), including W-MSE and Shuffled-DBN.

We also observe that CW-RGP obtains better performance compared to other non-contrastive methods like SimSiam and VICReg. We believe that a key advantage of CW-RGP over SimSiam is its guaranteed collapse avoidance, contributing to its superior performance compared to SimSiam. It has been shown in [5] and confirmed in our experiments that SimSiam sometimes suffers from collapse during training. The primary reason for the superiority of CW-RGP over VICReg lies in the full-rank constraint introduced by CW-RGP, which proves more effective in representation learning than the whitening constraint introduced by VICReg, as illustrated in Section V-B.

TABLE I
COMPARISON OF DIFFERENT SSL METHODS

Method	CIFAR-10		CIFAR-100		STL-10		Tiny-ImageNet	
	linear	5-nn	linear	5-nn	linear	5-nn	linear	5-nn
SimCLR [3]	91.80	88.42	66.83	56.56	90.51	85.68	48.84	32.86
BYOL [4]	91.73	89.45	66.60	56.82	91.99	88.64	**51.00**	**36.24**
SimSiam [5] (repro.)	90.51	86.82	66.04	55.79	88.91	84.84	48.29	34.21
Shuffled-DBN [13] (repro.)	90.45	88.15	66.07	56.97	89.20	84.51	48.60	32.14
Barlow Twins [36] (repro.)	88.51	86.53	65.78	55.76	88.36	83.71	47.44	32.65
VICReg [15] (repro.)	90.32	88.41	66.45	56.78	90.78	85.72	48.71	33.35
Zero-ICL [39] (repro.)	88.12	86.64	61.91	53.47	86.35	82.51	46.25	32.74
W-MSE 2 [16]	91.55	89.69	66.10	56.69	90.36	87.10	48.20	34.16
W-MSE 4 [16]	91.99	89.87	67.64	56.45	91.75	88.59	49.22	35.44
CW-RGP 2 (ours)	91.92	89.54	67.51	57.35	90.76	87.34	49.23	34.04
CW-RGP 4 (ours)	**92.47**	**90.74**	**68.26**	**58.67**	**92.04**	**88.95**	50.24	35.99

We use the same experimental setups as W-MSE [16]. We evaluate the classification accuracy (top 1) of a linear classifier and a 5-nearest neighbors classifier on different datasets with a ResNet-18 encoder. CW-RGP 2 and CW-RGP 4 indicate our methods with 2 and 4 positive views extracted per image, respectively. We use 'repro.' to indicate that the results are reproduced by our implementation.

2) Evaluation on Large-Scale ImageNet: We conduct experiments on the large-scale ImageNet dataset. Our implementation is based on the source code from *SimSiam* [5][5]. Except for the hyper-parameters relating to CW-RGP itself, we use the same setups as SimSiam [5]:

Encoder and Projector: We use the ResNet-50 [53] as the encoder, and the encoding dimension is 2048. We use a 3-layer MLP as the projector: two hidden layers with BN and ReLU applied to it and a linear layer as output. The hidden layer and embedding dimensions are 2048 and 1024, respectively.

Image Transformation Details: We use the same transformations in [5]: crop size from 0.2 to 1.0, no strengthened jittering (0.4, 0.4, 0.4, 0.1) with probability 0.8, grayscaling probability 0.2, and Gaussian blurring with 0.5 probability. We use standard protocols for performance evaluation [5].

Optimizer and Learning Rate Schedule: We apply the SGD optimizer, using a learning rate of $lr \times$ BatchSize / 256 with a base lr of 0.05 and cosine decay schedule. The weight decay is 10^{-4}, and the SGD momentum is 0.9. In addition, we use learning rate warm-up for the first 500 iterations of the optimizer.

We only experiment with the batch size of 256 and 512 due to memory limitations.

Evaluation Protocol: We use the same valuation protocol as in *SimSiam* [5]: Training the *linear classifier* for 100 epochs with the *LARS* optimizer (using a learning rate of $lr \times$ BatchSize / 256 with a base lr of 0.1 and cosine decay schedule). The batch size for evaluation is 1024.

Table II shows the results reported in [5], [16] and our findings using the code from BYOL [4], SwAV [29], and W-MSE 4 [16] with a batch size of 512 and the same training and evaluation settings as in [5]. CW-RGP 4 is trained with a batch size of 512 and achieves the highest accuracy among all methods under both 100 and 200 training epochs. CW-RGP also performs well when combined with the whitening penalty used in VICReg. Note that we also use a batch size of 256 under 100-epoch training, which obtains the top-1 accuracy of 69.5%.

3) Ablation Studies: Random Group Partition: We also conduct experiments to evaluate the effect of random group partition for channel whitening. We use 'CW', 'CW-GP', and 'CW-RGP' to indicate channel whitening without group partition, with group partition, and with random group partition, respectively. We consider the setups with $s = 2$ and $s = 4$ positive views and use the same setup as in Table I. The results in Table IV show that random group partition facilitates channel whitening in obtaining better results. Fig. 6 also shows that CW with random group partition helps improve the performance.

[5][Online]. Available: https://github.com/facebookresearch/simsiam.

TABLE II
COMPARISONS ON IMAGENET LINEAR CLASSIFICATION

Method	Batch size	100 eps	200 eps
SimCLR [3]	4096	66.5	68.3
MoCo v2 [25]	256	67.4	69.9
BYOL [4]	4096	66.5	70.6
SwAV [29]	4096	66.5	69.1
SimSiam [5]	256	68.1	70.0
W-MSE 4 [16]	4096	69.4	-
Zero-CL [39]	1024	68.9	-
BYOL [4] (repro.)	512	66.1	69.2
SwAV [29] (repro.)	512	65.8	67.9
W-MSE 4 [16] (repro.)	512	66.7	67.9
CW-RGP 4 (ours)	512	**69.7**	**71.0**

We use the same setups as SimSiam [5]. All results are based on the ResNet-50 encoder. Some results are reported directly from [5], and some are reproduced by our implementation (denoted by 'repro.'). CW-RGP 4 indicates our methods with 4 positive views extracted per image.

Batch Size: We conduct experiments to evaluate CW against BW in terms of stability using different batch sizes. We train CW and BW on the ImageNet-100 dataset, using batch sizes ranging from $\{32, 64, 128, 256\}$. Fig. 9 shows that CW performs more robustly when using small batches for training.

B. Transfer to Downstream Tasks

We examine the representation strength by transferring our model to other tasks, including object detection using the VOC [54] and COCO [55] datasets. In addition, we evaluate our method on instance segmentation using the COCO [55] dataset. We use the baseline (except for the pre-trained model, all the other components are the same) of the detection codebase from MoCo [2] for CW-RGP. We use the default hyper-parameter settings from the codebase for CW-RGP, using our 200-epoch pre-trained model on ImageNet. For the experiments on object

TABLE III
TRANSFER TO OBJECT DETECTION AND INSTANCE SEGMENTATION

Method	VOC 07+12 detection			COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}
Scratch	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
IN-supervised	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR [3]	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 [25]	**82.3**	57.0	63.3	58.8	39.2	42.5	55.5	34.3	36.6
BYOL [4]	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV [29]	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam [5]	82.0	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
CW-RGP (ours)	$82.2_{\pm 0.07}$	**57.2**$_{\pm 0.10}$	**63.8**$_{\pm 0.11}$	**60.5**$_{\pm 0.28}$	**40.7**$_{\pm 0.14}$	**44.1**$_{\pm 0.14}$	**57.3**$_{\pm 0.16}$	**35.5**$_{\pm 0.12}$	**37.9**$_{\pm 0.14}$

We follow the experimental setups in [5]. We use Faster R-CNN and Mask R-CNN for VOC and COCO datasets, respectively. The backbone is the ResNet-50 pre-trained on ImageNet over 200 epochs using different SSL methods for comparison. The table is mostly inherited from [5]. Our CW-RGP is performed with three random seeds, with mean and standard deviation reported.

TABLE IV
RESULTS OF ABLATION FOR RANDOM GROUP PARTITION

Method	CIFAR-10		CIFAR-100	
	linear	5-nn	linear	5-nn
CW 2	91.66	88.99	66.26	56.36
CW-GP 2	91.61	88.89	66.17	56.53
CW-RGP 2	**91.92**	**89.54**	**67.51**	**57.35**
CW 4	92.10	90.12	66.90	57.12
CW-GP 4	92.08	90.06	67.34	57.28
CW-RGP 4	**92.47**	**90.74**	**68.26**	**58.67**

We use the same setups as in Table 1. We use 'CW', 'CW-GP', and 'CW-RGP' to indicate channel whitening without group partition, with group partition, and with random group partition, respectively. 'CW s' indicates s positive views.

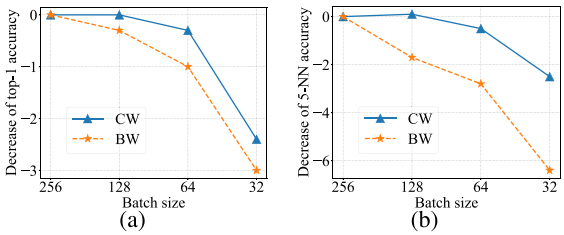


Fig. 9. Results using different batch sizes. We train CW and BW over 100 epochs on ImageNet-100. The x-axis is the batch size, while the y-axis is the decrease in performance compared to the batch size of 256. We evaluate the top-1 and 5-nn accuracy (%) in (a) and (b), respectively.

detection using the VOC dataset, we use Faster R-CNN which is fine-tuned on the VOC 2007 trainval and 2012 train sets and evaluated on the VOC 2007 test set. For object detection and instance segmentation experiments on the CoCo dataset, we use Mask R-CNN ($1\times$ schedule) which is fine-tuned on the COCO 2017 train set and evaluated on the COCO 2017 val set. All Faster/Mask R-CNN models are with the C4-backbone.

The experiments on CW-RGP are carried out with 3 random seeds and the mean performance and standard deviation are reported. The baseline results shown in Table III are reported in [5]. Overall, CW-RGP performs better than or on par with these state-of-the-art approaches on COCO object detection and instance segmentation, which shows the potential of CW-RGP in transferring to downstream tasks.

IX. CONCLUSION

In this paper, we study whitening loss for SSL and observe several interesting results. We show that batch whitening (BW) based methods only require the embedding to be full-rank, which is also a sufficient condition for collapse avoidance. We also theoretically demonstrate that the stable rank of the embedding is invariant during the training by gradient descent with an infinitely small learning rate. Motivated by the theoretical justification, we propose channel whitening with random group partition (CW-RGP) and empirically demonstrate its effectiveness against the state-of-the-art approaches on benchmark datasets.

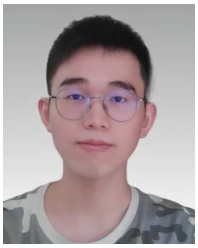
REFERENCES

[1] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 1392.
[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.
[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, Art. no. 149.
[4] J.-B. Grill et al., "Bootstrap your own latent-A new approach to self-supervised learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 1786.
[5] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15750–15758.
[6] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.
[7] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2412–2429, Feb. 2023.
[8] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," 2020, *arXiv: 2011.00362*.
[9] K. Ranasinghe, M. Naseer, S. Khan, F. S. Khan, and M. Ryoo, "Self-supervised video transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2864–2874.
[10] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.
[11] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5628–5637.
[12] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, "Boosting contrastive self-supervised learning with false negative cancellation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2785–2795.

[13] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao, "On feature decorrelation in self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9578–9588.
[14] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022.
[15] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022.
[16] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, "Whitening for self-supervised representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3015–3024.
[17] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 791–800.
[18] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," *Amer. Statistician*, vol. 72, no. 4, pp. 309–314, 2018.
[19] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*. Cambridge, U.K.: Cambridge Univ. Press, 2018.
[20] X. Weng, L. Huang, L. Zhao, R. Anwer, S. H. Khan, and F. Shahbaz Khan, "An investigation into whitening loss for self-supervised learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 29748–29760.
[21] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6210–6219.
[22] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv: 1807.03748*.
[23] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4182–4192.
[24] H. B. Barlow et al., "Possible principles underlying the transformation of sensory messages," *Sensory Commun.*, vol. 1, no. 1, pp. 217–233, 1961.
[25] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv: 2003.04297*.
[26] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9620–9629.
[27] C. Li et al., "Efficient self-supervised vision transformers for representation learning," in *Proc. Int. Conf. Learn. Representations*, 2022.
[28] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 139–156.
[29] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.
[30] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Representations*, 2021.
[31] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9630–9640.
[32] C. Zhang, K. Zhang, C. Zhang, T. X. Pham, C. D. Yoo, and I. S. Kweon, "How does simsiam avoid collapse without negative samples? A unified understanding with self-supervised contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2022.
[33] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10268–10278.
[34] C. Tao et al., "Exploring the equivalence of Siamese self-supervised learning via A unified gradient framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14411–14420.
[35] Q. Garrido, Y. Chen, A. Bardes, L. Najman, and Y. LeCun, "On the duality between contrastive and non-contrastive self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2023.
[36] J. Zbontar, L. Jing, I. Misra, Y. Lecun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12310–12320.
[37] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 76–89.
[38] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10173–10196, Aug. 2023.
[39] S. Zhang, F. Zhu, J. Yan, R. Zhao, and X. Yang, "Zero-CL: Instance and feature decorrelation for negative-free symmetric contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2022.
[40] B. He and M. Ozay, "Exploring the gap between collapsed and whitened features in self-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8613–8634.
[41] K. K. Agrawal, A. K. Mondal, A. Ghosh, and B. Richards, "α-ReQ: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 17626–17638.
[42] Q. Garrido, R. Balestriero, L. Najman, and Y. Lecun, "RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 10929–10974.
[43] A. Siarohin, E. Sangineto, and N. Sebe, "Whitening and coloring batch transform for GANs," in *Proc. Int. Conf. Learn. Representations*, 2019.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[45] L. Huang, L. Zhao, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "An investigation into the stochasticity of batch whitening," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6438–6447.
[46] J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma, "Provable guarantees for self-supervised deep learning with spectral contrastive loss," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 5000–5011.
[47] C. Ye et al., "Network deconvolution," in *Proc. Int. Conf. Learn. Representations*, 2020.
[48] W. Wang, Z. Dang, Y. Hu, P. Fua, and M. Salzmann, "Robust differentiable SVD," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5472–5487, Sep. 2022.
[49] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep. TR-2009, 2009.
[50] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[51] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, 2015.
[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[54] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–308, Sep. 2009.
[55] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

Lei Huang received the BSc and PhD degrees from the School of Computer Science and Engineering, Beihang University. He is currently an associate professor with the Institute of Artificial Intelligence, Beihang University, China. From 2015 to 2016, he visited the Vision and Learning Lab, University of Michigan, Ann Arbor, USA. During 2018 to 2020, he was a research scientist in Inception Institute of Artificial Intelligence (IIAI), UAE. He has published more than 50 papers in high-impact scientific journals and conferences. He has been awarded the best paper nomination in CVPR 2020. His research interests include machine learning and computers vision.

Yunhao Ni is currently working toward the graduate degree with the Institute of Artificial Intelligence of Beihang University. His main research direction is applied mathematics and machine learning.



Xi Weng is currently working toward the graduate degree with the Institute of Artificial Intelligence of Beihang University. His main research direction is self-supervised learning.

Rao Muhammad Anwer received the MSc degree in intelligent systems design from the Chalmers University of Technology, Sweden, and the PhD degree in computer vision from the Autonomous University of Barcelona, Spain. He is an assistant professor with the MBZ University of Artificial Intelligence. Prior to joining MBZUAI, he was with the Inception Institute of Artificial Intelligence (IIAI) in Abu Dhabi working as research scientist from 2018 to 2020. Before joining IIAI, he worked as a post doctoral research fellow with Aalto University, Finland from 2014 to 2018. His research interests are in object detection, instant segmentation, human object interaction, and pedestrian detection.

Salman Khan received the PhD degree from the University of Western Australia, in 2016. He is an associate professor with the MBZ University of Artificial Intelligence. He has been an honorary faculty with the Australian National University since 2016. His thesis received an honorable mention on the Dean's List Award. He has been awarded the Outstanding Reviewer award at IEEE CVPR multiple times, won the Best Paper award at 9th ICPRAM 2020, and won 2nd prize in the NTIRE Image Enhancement Competition alongside CVPR 2019. He has published more than 100 papers in high-impact scientific journals and conferences. His research interests include computer vision and machine learning. He served as an area chair member for several premier conferences, including IEEE/CVF CVPR, ICCV, ICML, ECCV, ACCV, and NeurIPS. He co-organized special issues in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, *ACM Transactions on Multimedia Computing* and *MDPI Remote Sensing journals*. He co-organized workshops at CVPR, ICCV, ACCV, and NeurIPS and will be a tutorial co-chair at ICCV. His research interests include computer vision and machine learning.

Ming-Hsuan Yang (Fellow, IEEE) is affiliated with Google, UC Merced, and Yonsei University. Yang serves as a program co-chair of IEEE International Conference on Computer Vision (ICCV) in 2019, program co-chair of the Asian Conference on Computer Vision (ACCV) in 2014, and general co-chair of ACCV 2016. He served as an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* and is an associate editor of the *International Journal of Computer Vision, Image and Vision Computing* and *Journal of Artificial Intelligence Research*. He received the NSF CAREER award and Google Faculty Award.

Fahad Shahbaz Khan received the MSc degree in intelligent systems design from the Chalmers University of Technology, Sweden, and the PhD degree in computer vision from the Autonomous University of Barcelona, Spain. He is currently a full professor and deputy department chair of computer vision with the MBZUAI, Abu Dhabi, United Arab Emirates. He also holds a faculty position (Universitetslektor + Docent) with Computer Vision Laboratory, Linkoping University, Sweden. From 2018 to 2020 he worked as a lead scientist with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. He has achieved top ranks on various international challenges (Visual Object Tracking VOT: 1st 2014 and 2018, 2nd 2015, 1st 2016; VOT-TIR: 1st 2015 and 2016; OpenCV Tracking: 1st 2015; 1st PASCAL VOC 2010). His research interests include a wide range of topics within computer vision and machine learning, such as object recognition, object detection, action recognition, multimodal learning and visual tracking. He has published articles in high impact computer vision journals and conferences in these areas. He regularly serves as a program committee member for leading computer vision conferences such as CVPR, ICCV, and ECCV.