An Embedding-Dynamic Approach to Self-Supervised Learning

Suhong Moon
UC Berkeley
suhong.moon@berkeley.edu

Domas Buracas
UC Berkeley
dominykas@berkeley.edu

Seunghyun Park
Clova AI Research, NAVER Corp
seung.park@navercorp.com

Jinkyu Kim
Korea University
jinkyukim@korea.ac.kr

John Canny
UC Berkeley
canny@berkeley.edu

Abstract

*A number of recent self-supervised learning methods have shown impressive performance on image classification and other tasks. A somewhat bewildering variety of techniques have been used, not always with a clear understanding of the reasons for their benefits, especially when used in combination. Here we treat the embeddings of images as point particles and consider model optimization as a dynamic process on this system of particles. Our dynamic model combines an attractive force for similar images, a locally dispersive force to avoid local collapse, and a global dispersive force to achieve a globally-homogeneous distribution of particles. The dynamic perspective highlights the advantage of using a delayed-parameter image embedding (a la BYOL) together with multiple views of the same image. It also uses a purely-dynamic local dispersive force (Brownian motion) that shows improved performance over other methods and does not require knowledge of other particle coordinates. The method is called MSBReg which stands for (i) a **M**ultiview centroid loss, which applies an attractive force to pull different image view embeddings toward their centroid, (ii) a **S**ingular value loss, which pushes the particle system toward spatially homogeneous density, (iii) a **B**rownian diffusive loss. We evaluate downstream classification performance of MSBReg on ImageNet as well as transfer learning tasks including fine-grained classification, multi-class object classification, object detection, and instance segmentation. In addition, we also show that applying our regularization term to other methods further improves their performance and stabilize the training by preventing a mode collapse.*

1. Introduction

A good representation should include useful features (those that facilitate downstream prediction tasks) while ignoring "nuisance" features [3]. Among the best known self-supervised methods, contrastive methods combine an attractive term between similar images (typically different perturbations of the same image) with an explicit repulsive term between distinct pairs. Recently, BYOL [14] utilized siamese neural networks (referred to as the online and target) with lagged (moving averaged) parameters in the target network, and simply minimized distance between online and target network embeddings. While there was no *explicit* repulsive term in BYOL, it was later shown to be highly dependent on the use of BatchNorm layers. The activation normalization in BatchNorm and other methods can be viewed as a global, dimension-wise dispersion of the set of embeddings, a desirable feature of a representation. However, other normalization methods such as LayerNorm were shown to be much less effective in BYOL suggesting the story is more complicated than normalization and global dispersion [1, 21, 11, 23].

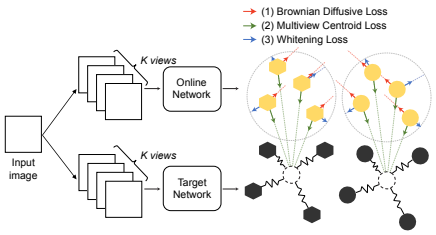

Figure 1. Our proposed MSBReg for SSL contains the following three regularization terms. (1) A Brownian diffusive loss (red), which induces a random motion of embeddings. This provides an implicit contrastive effect, preventing collapse of emebddings and stretching weaker links more on average than strong ones. (2) A Multiview centroid loss (green), where we train our online network to minimize the distance between centroids of online and target network embeddings of different views of the same image. (3) A Singular value loss (blue), which decorrelates the different feature dimensions to disperse embeddings uniformly in the embedding space. Positive pairs are indicated with the same shapes.

Inspection of the gradients in BatchNorm reveals that

they have a strong stochastic component (beyond the global "normalizing" component) that depends on differences between image activations and their batch centroid (i.e. on whatever other images happen to be in the same minibatch). From our perspective, these forces provide a local (stochastic) dispersive force between embeddings. Thus Batchnorm implements two of the desirable features of a good representation (local and global dispersion of embeddings), but in a suboptimal way. Here we define separate loss terms for local and global dispersion and apply them to the embedding layer only (as opposed to other intermediate network layers). By moving dispersion to loss layers, we allow network normalization layers (which ideally impact network training and stability but not losses) to be independently designed. We can also separately define and optimize the local and global dispersion losses.

If we assume that the optimization method used to train the network is either stateless or sufficiently "fast" (e.g. the optimizer uses momentum=0.9 for an effective time constant of 10 steps), then the embeddings are part of a second-order dynamical system. The embeddings are defined by the parameters of the two networks (online and target), together with corresponding input images. The moving parameter average implemented on the target network, together with a fast optimizer which functions as an integrator of loss gradients, defines a second-order dynamical system. We exploit the dynamics of this system in two ways: by using "fast-slow" optimization for attractive and dispersive forces, and by showing that stochastic energy injected into the system should "stretch" attractive links with equal potential energy on average - so weaker attractive links will be stretched further.

Multiview contrastive training, where more than two augmentations are compared, works very effectively with a lagged-arm network. Dispersive forces act in the network to situate embeddings with globally uniform density. While loss-gradients act as forces applied to the online network, online embeddings experience a strong viscous "drag" from their corresponding target network embeddings which they are attracted to. So embeddings move globally at the time constant of the lagged network, which is typically thousands to tens of thousands of time steps. On the other hand, embeddings within the same group, i.e. embeddings of views of the same image, experience no "drag" relative to their centroid. They collapse and are maintained close together at the time constant of the online network.

Given a lagged-arm, siamese architecture inspired by BYOL, we explore Multiview, Singular value, Brownian Diffusive regularizations. These three loss terms address respectively, (i) fast-slow attractive/dispersive optimization, (ii) global, uniform, dispersion of embeddings (iii) local dispersive force.

We evaluate our approach on visual benchmarks including ImageNet-100 [8], STL-10 [7], and ImageNet [8]. We show that our model significantly outperforms prior work in the image classification task. Also, we show that joint local/global dispersive forces lead to a larger dissimilarity of negative pairs compared to other approaches. We summarize our contributions as follows:

- We analyze and optimize self-supervised learning using a dynamic model of embeddings.
- To optimize the placement of embeddings, we propose a MSBReg loss that consists of 1) Multiview centroid loss and 3) Singular value loss 3) Brownian diffusive loss and MSBReg outperforms other baselines by a significant margin.

2. Related Work

Self Supervised Learning. Recent works suggest that a state-of-the-art image representation can be successfully and efficiently learned by a discriminative approach with self supervised learning. These methods originally relied on a contrastive loss that encourages representation of different views of the same image (i.e. positive pairs) to be close in the embedding space, and representations of views from different images (i.e. negative pairs) to be pushed away from each other. Contrastive methods often require a careful treatment of negative pairs, which need a large memory overhead as they need to be sampled from a memory bank [27, 15] or from the current mini-batch of data [5]. The contrastive approach is also unsatisfying from a modeling perspective - the fact that images are distinct does not imply that they are different - but the contrastive approad applies large repulsive gradients to distinct, close image pairs.

Motivated by a desire to overcome the difficulties of contrastive approaches, recent works [14, 6] use two neural networks (referred to as online and target networks) are trained to minimize the distance between their embeddings of the same image. Some works use a moving average on parameters of one arm [14] while others use the same parameters [15, 6]. These methods have been effective but their success is somewhat mysterious since there is no obvious force to prevent collapse of embeddings since forces are only attractive. It turn out that as batch normalization [17] was an important element of the success of BYOL. In contrast we employ explicit local and global dispersive losses in addition to attractive forces on groups of multiple image views.

Regularizing Consistency of Singular Value. Whitening is the most similar approach to regularizing consistency of singular values. Recently, whitening output embeddings has received attention as a method to avoid a mode collapse. Whitening removes redundant information in input and prevents all dimensions of embeddings from being encoded

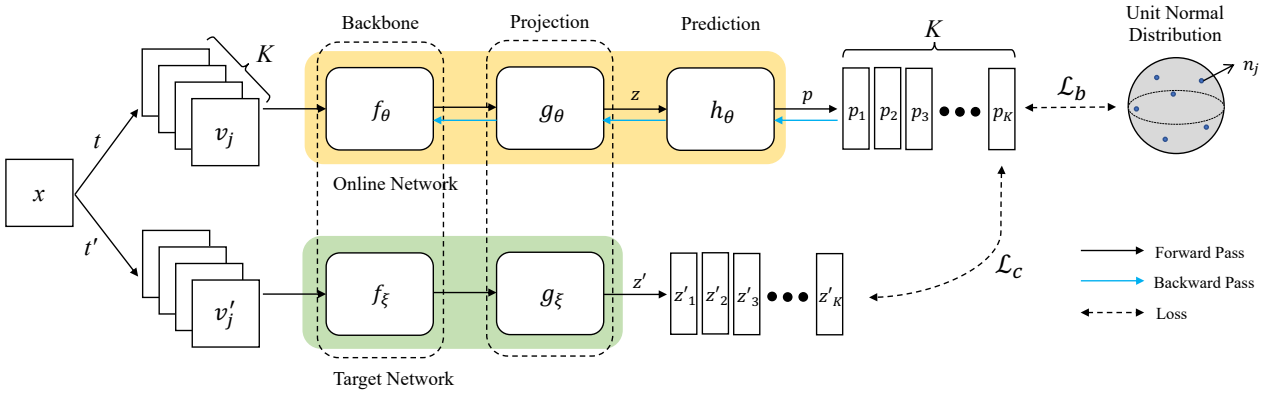

Figure 2. The architecture overview of MSBReg . This architecture is inspired by BYOL's architecture. Each model takes K augmented views as its inputs. MSBReg minimizes (1) multiview centroid loss, (2) singular value loss, (3) Brownian diffusive loss and . The first makes the online network predict the target network's representation of the centroid of K views. The second loss favors a spatially uniform (spherical) distribution. The last one induces noise into embedding space and makes the embeddings repulse each other on average, preventing the model from converging to collapsed solutions.

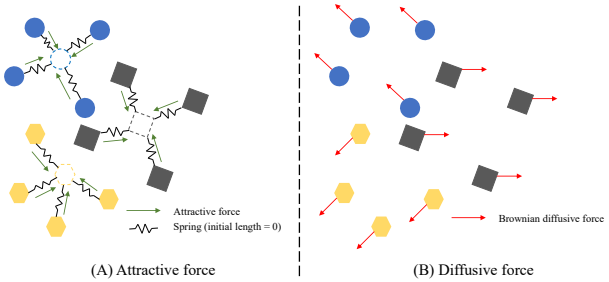

Figure 3. (A) Multiview centroid loss applies attractive force to embeddings generated by online network. The solid shapes are embeddings generated by online network and the shapes with dashed line are the geometric centroids of the embeddings generated by target network. We can model such system as spring-mass system. (B) Brownian diffusive loss induces the random walk of embeddings, preventing the model from admitting collapsed solutions.

with the same representation. Whitening features induces the contrastive effect of the embeddings by scattering them. This [9] performs a explicit whitening via Cholesky decomposition. Performing Cholesky decomposition, which requires the computation of inverse of the sample covariance matrix is not only computationally expensive but unstable. This method [29] computes cross-correlation matrix and makes it close to identity matrix in Frobenius norm. This paper [2] suggested a similar approach. Unlike the methods mentioned above, which compute the covariance matrix with the only positive pairs, singular value loss term in MSBReg computes the covariance matrix along the batch dimension (with the negative pairs) and helps global dispersion of embeddings by making the embeddings be distributed isotropically. To emphasize this aspect, we coinage our loss as singular value loss, which regularizes the consistency of singular values of the empirical covariance matrix.

Multiview Loss. In supervised learning settings, batch repetition method [16] is proposed to improve the image classification performance as well as training efficiency. Recent self-supervised learning based on contrastive learning usually uses two views of the same image as positive pairs. And it is trained to minimize the distance or maximize similarity of embeddings of those two views. Recent work [4] suggested multi-crop method which maximizes the similarity between views more than 2. To reduce computational cost, it generates 2 views with high resolution (224×224 for ImageNet) and several other views with low resolution (96×96 for ImageNet). This method [9] generates multiple positive views to perform whitening among them. In contrast, our method uses multiple views of the positive views of the same images to compute the centroid and distance between embeddings and the centroid. We discuss the relationship between batch repetition method and multiview centroid loss in the appendix.

Uniformity of Embeddings. Our work is aligned with [25] in that our method also seeks to distribute the embeddings as uniformly as possible on the embedding space. That work claims that the contrastive learning is to make embeddings be distributed uniformly on the hypersphere. Similarly, our work also tries to distribute the embeddings uniformly on the embedding space. That method reformulates contrastive loss as the sum of alignment loss and uniformity loss. The first term, alignment loss, aligns positive views. The second term, uniformity loss, makes the embedding distribution *uniform* on the surface of unit sphere. The uniformity loss is defined by Gaussian kernel. The difference to our method is that 1) [25] is based on constrastive method and 2) [25] hypothesizes the embedding space is hypersphere. However, our method seeks more general embedding space with the dynamical system modeling. The advantage of

modeling dynamical system is that we can study the motion of embeddings and control them with this model. We further study the effect of our loss terms to uniformity and alignment trade-off in depth in the appendix.

3. Method

3.1. BYOL Architecture

We follow the recent BYOL architecture [14] that learns a joint embedding of an image $x \in \mathcal{X}$ with two networks – consists of two neural networks referred to as the *online* (or fast learner) and *target* (or slow learner) network. For completeness, we summarize some of the key details of the BYOL architecture. As shown in Figure 2, the online network is trained to predict the target network's representation of the augmented view of the same image. This online network is parameterized by a set of learnable weights θ and consists of three consecutive components: a backbone f_θ, a projection head g_θ, and a prediction head h_θ. The target network is parameterized by a set of weights ξ and consists of two components: a backbone f_ξ and a projection head g_ξ. The parameter ξ is updated by the bias-corrected exponentially weighted moving average of the online network's parameter θ at each training step, i.e. $\xi_{t+1} = \tau_t \xi_t + (1-\tau_t)\theta_t$ where $\tau_t \in [0,1]$ is a target decay rate.

Two augmented views $v \triangleq t(x)$ and $v' \triangleq t'(x)$ are generated by applying image augmentations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$ given two distributions of image augmentations $\mathcal{T}$ and $\mathcal{T}'$. The online network outputs $z \triangleq g_\theta(f_\theta(v))$ from the first augmented view v, while the target network produces $z' \triangleq g_\xi(f_\xi(v'))$ from the second augmented view v'. A prediction from the online network $p \triangleq h_\theta(z)$ is then l_2-normalized to compute the cosine similarity loss $\mathcal{L}_{\text{byol}}$ by measuring mean squared error between the normalized prediction p and the normalized target predictions z':

$$\mathcal{L}_{\text{byol}}(\theta, \xi; \mathcal{X}) := \|\hat{p} - \hat{z}'\|_2^2 = 2 - 2\frac{\langle p, z' \rangle}{||p||_2 \cdot ||z'||_2} \quad (1)$$

where $\hat{p} = p/\|p\|_2$ and $\hat{z}' = z'/\|z'\|_2$. Note that the loss $\mathcal{L}_{\text{byol}}$ is optimized with respect to θ only, but not ξ. The gradient does not back-propagate through the target network as depicted by *stop-gradient* in Figure 2. After training, both the prediction head h_θ and the projection head g_θ are discarded and the representations z of the online network are used for downstream tasks.

3.2. MSBReg

Built upon BYOL architecture, we use the following loss $\mathcal{L}(\theta, \xi; \mathcal{X})$ (instead of using $\mathcal{L}_{\text{byol}}$) that consists of the following three loss terms: (i) multiview centroid loss $\mathcal{L}_c$, (ii) singular value loss $\mathcal{L}_s$, and (iii) Brownian diffusion loss $\mathcal{L}_b$. The overall loss is defined as follows:

$$\mathcal{L}(\theta, \xi; \mathcal{X}) = \mathcal{L}_c(\theta, \xi; \mathcal{X}) + \lambda_s \mathcal{L}_s(\theta; \mathcal{X}) + \lambda_b \mathcal{L}_b(\theta; \mathcal{X}) \quad (2)$$

Multiview Centroid Loss. As opposed to BYOL, we train the online network to predict the target network's *centroid* representation of differently augmented multi-views of the same image. Given an image $x \in \mathcal{X}$, we generate K differently augmented views (i.e. multi-view): $v_j \triangleq t_j(x)$ and $v_l \triangleq t_l(x)$ for $j,\ l \in \{1, 2, \ldots, K\}$ by applying stochastic image augmentations $t_j, t_l \sim \mathcal{T}$. Given K outputs from the target network, $z'_l = g_\xi(f_\xi(v'_l))$, we use the geometric center of these K outputs as the centroid representation, i.e. $\frac{1}{K}\sum_{l=1}^{K} \hat{z}'_l$ where $\hat{z}'_l = z'_l/\|z'_l\|_2$. Lastly, we compute the sum of L_2 loss between the target network's centroid representation of embeddings of K different views $\{\hat{z}'_l\}_1^K$ and the online network's representation – thus, this loss applies an attractive force to pull together multiple augmented representations of the same image (positive pairs) into the geometric centroid as the pivot to cluster embeddings.

Ultimately, we the following multiview centroid loss $\mathcal{L}_c$:

$$\mathcal{L}_c(\theta, \xi; \mathcal{X}) = \frac{1}{K}\sum_{j=1}^{K} \left\| \hat{p}_j - \frac{1}{K}\sum_{l=1}^{K} \hat{z}'_l \right\|_2^2 \quad (3)$$

where $\hat{p}_j = p_j/\|p_j\|_2$ is l_2-normalized predictions from the online network for the augmented view of the same input, i.e. $p_j = h_\theta(g_\theta(f_\theta(v_j)))$. Note that, minimizing Eq. 3 is mathematically identical to minimizing pairwise distance between $\{\hat{p}'_j\}_1^K$ and $\{\hat{z}'_l\}_1^K$. Therefore, this loss generates a stronger attractive force that aggregates the embeddings of the same image that BYOL loss in Eq. 1.

Brownian Diffusion Loss. We use a dispersive loss, called Brownian diffusion loss, that induces a Brownian motion (or a random walk) of the online network's representation p_j of j-th augmented view of an input. A d-dimensional random vector $n \in \mathbb{R}^d$ is sampled from unit normal distribution, i.e. $n \sim \mathcal{N}(0, I_d)$ with an identity matrix I_d. Our Brownian diffusion loss is defined as follows:

$$\mathcal{L}_b(\theta; \mathcal{X}) = \frac{1}{K}\sum_{j=1}^{K} \langle \hat{n}, \hat{p}_j \rangle \quad (4)$$

where $\hat{n} = n/\|n\|_2$. The noise vector $\hat{n}$ drives a diffusive motion by pushing particles in the embedding space in radial direction, which is uniformly sampled on the unit hyper-sphere.

Importantly, we use the same random vector $\hat{n}$ for the all augmented embeddings of the given image. This implies that the positive pairs which share the similar semantics are not spread apart. In contrast, the views from the different

image moves to the different direction and the direction is likely to be orthogonal to other images' moving direction. I.e. Brownian diffusion loss disperses the embeddings locally, which gives implicit contrastive effect between embeddings of different images.

We observe that our Brownian diffusion loss is critical to prevent mode-collapse [14]. As the target network's parameter is updated by the exponentially weighted moving average of the online network's parameter at each training step (given a high target decay rate), the change of the target network's representation is relatively slower than that of the online network (effectively $\frac{1}{1-\tau}$ times slower). Such an imbalance may cause a mode collapse as the online network's representation can quickly collapse into a single point without any repulsive force between them.

Singular Value Loss. Lastly, we use the singular value loss $\mathcal{L}_w$ that decorrelates the different feature dimensions of the projections $\hat{p}$ to prevent these dimensions from conveying the same information, thus avoid a dimension collapse. We minimize the following Euclidean distance between the empirical covariance matrix of the embeddings and the identity matrix I_d – thus, we penalize the off-diagonal coefficients of the covariance matrix and make the distribution ball-shaped. Let the p_{ij} be i-th batch and j-th augmented embeddings. Then the empirical covariance matrix of j-th augmented embeddings S_j is:

$$S_j = \frac{1}{n-1}\sum_{i=1}^{n}(p_{ij} - \bar{p_j})(p_{ij} - \bar{p_j})^T \tag{5}$$

where n is the number of batches and $\bar{p_j} = \frac{1}{n}\sum_{i=1}^{n} p_{ij}$. Then we define singular value loss as:

$$\mathcal{L}_s(\theta; \mathcal{X}) = \frac{1}{K}\sum_{j=1}^{K} \|S_j - I_d\|_{\mathrm{F}}^2 \tag{6}$$

$$= \frac{1}{K}\sum_{j=1}^{K}\sum_{i=1}^{d}(\sigma_{ij} - 1)^2 \tag{7}$$

where σ_{ij} is singular values of the covariance matrix, S_j. We found that this loss improves when corporated with Brownian diffusion loss.

Some prior works [2, 9, 29] justify whitening loss as removing correlations between different embeddings. In our method however, we treat singular value loss as a dispersive force that encourages uniformity of the embedding distribution. Even though Brownian diffusion loss addresses local dispersion in the embedding space, singular value loss exerts the force to regularize the shape of embedding distribution to be globally spherical at large scale.

4. Experiments

4.1. Evaluation of Representations with MSBReg

Evaluation on ImageNet-100 and STL-10. Following the linear evaluation protocol, we train a simple linear classifier with the frozen representations from our encoder, which is pre-trained with our MSBReg . We first evaluate the performance of the encoder on a small-size ImageNet-100 [22] and a mid-size STL-10 datasets. We observe in Table 1 that the performance of MSBReg generally outperforms other state-of-the-art approaches on both datasets, especially we observe a large gain on the STL-10 dataset. The performance gain is more apparent that the following three approaches, MoCo, SimCLR, and Wang and Isola, use a more expressive ResNet-50-based backbone than our ResNet-18-based backbone. We also observe that the quality of the learned representation improves as the number of views K increases (compare bottom two rows).

Evaluation on ImageNet. We further evaluate the representations obtained after self-supervised pre-training with MSBReg on the large-scale ImageNet dataset with two evaluation metrics: 1) linear evaluation protocol and 2) semi-supervised learning with the subsets of ImageNet and 3) kNN classification. For linear evaluation protocol, likewise to Table 1, dashed-line means that the original paper did not report the corresponding value. We observe in Table 2 that the performance of MSBReg outperforms other approaches and gets the result compatible to SwAV with multi-crop, which may confirm that the effectiveness of MSBReg for learning the better visual representation. Especially, compared to other baselines which are trained for 400 epochs, our method is only trained for 300 epochs. This implies that it has enough room to improve a lot. Again, note that ours uses a smaller batch size than alternatives except MoCo-v2 (i.e. 512 vs. 4096 or 1024), but shows the matched or better performance.

To evaluate semi-supervised leaning ability of our method, we report top-1 and top-5 accuracy over 1% and 10% of ImageNet subsets. The experiment results are in Table 3. For both 1% and 10% subsets, our method outperforms baselines, when we compare methods with top-1 accuracy. Especially, in the fine-tuning result with 1% subset of ImageNet dataset (see 1st column in Table 3), our method surpasses with the large margin. For top-5 accuracy, our method gets matched performance with [13] and outperforms other methods.

kNN evaluation results are in Table 4. We report the accuracy of 20-NN and 200-NN classification results. Our method outperforms baselines.

Table 1. Classification accuracy (top-1 and top-5) of a linear classifier and 5-nearest neighbors (5-NN) classifier for different loss functions on two visual benchmarks: ImageNet-100 [8] and STL-10 [7]. Note that BYOL (1st row), W-MSE 4 (5th row), and ours (the bottom 2 rows) are based on a ResNet-18 encoder, while others on a more expressive ResNet-50 encoder. †: scores are from our reproduction.

Method	Backbone	ImageNet-100 [22]			STL-10 [7]	
		Top-1 (%)	Top-5 (%)	5-NN (%)	Top-1 (%)	5-NN (%)
BYOL† [14]	ResNet-18	71.56	91.18	63.18	89.50	85.15
MoCo [15]	ResNet-50	72.80	91.64	-	-	-
SimCLR [5]	ResNet-50	-	-	-	90.51	85.68
Wang and Isola [25]	ResNet-50	74.60	92.74	-	-	-
W-MSE 4 [9]	ResNet-18	79.02	94.46	71.32	91.75	88.59
Ours ($K = 4$)	ResNet-18	80.38	94.92	74.30	93.00	90.38
Ours ($K = 8$)	ResNet-18	**81.56**	**95.20**	**75.24**	**93.19**	**90.56**

Table 2. Downstream task result comparison on ImageNet. The backbone architecture for all the methods is ResNet-50. Note that the baseline results are from [6] and [9]. Bold face is the best accuracy and the underline is the second best accuracy. † means that our model is trained 300 epochs.

Method	Batch Size	Epoch		
		100	200	400
BYOL [14]	4,096	66.5	70.6	73.2
SimCLR [5]	4,096	66.5	68.3	69.8
MoCo-v2 [15]	256	67.4	69.9	71.0
W-MSE 4 [9]	1,024	69.3	-	72.56
SwAV [4] (w.o. multi-crop)	4,096	66.5	69.1	70.7
SwAV [4] (with multi-crop)	4,096	**72.1**	**73.9**	**74.6**
SimSiam [6]	256	68.1	70.0	70.8
Ours ($K = 4$)	512	<u>70.7</u>	<u>73.8</u>	**74.6**†

Table 3. Semi-Supervised classification result comparison on the subsets of ImageNet. We finetune the classifier and the encoder with 1% and 10% of labeled data of ImageNet. We report top-1 and top-5 accuracy. Bold face is the best accuracy.

Method	Top-1 (%)		Top-5 (%)	
	1%	10%	1%	10%
SimCLR [5]	48.3	65.6	75.5	87.8
BYOL [14]	53.2	68.8	78.4	89.0
VICReg [2]	54.8	69.5	79.4	89.5
SwAV [4] (with multi-crop)	53.9	70.2	78.5	89.9
Barlow Twins [29]	55.5	69.7	79.2	89.3
OBoW [13]	-	-	**82.9**	**90.7**
Ours ($K = 4$)	**58.6**	**70.6**	81.9	90.1

4.2. Transfer Learning on Various Downstream Tasks

We further evaluate the transferability of the features trained with MSBReg on ImageNet via transferring the features to various downstream tasks. In Table 5, we compare the performance of MSBReg with baselines. We first report the linear classification result on VOC07 [10], Places205 [30] and iNaturalist [24] visual benchmarks. Each of benchmarks is to evaluate 1) multi-label classification 2) scenic scenario and 3) fine-grained classifcation. We evalute the performance of linear classifier on top of the frozen ResNet-50 encoder pretrained with MSBReg method. We report mAP for VOC07 dataset and top-1 accuracy (%) for other benchmarks. We observe that our method shows generally matched results compared with alternatives.

Table 4. kNN classification result comparison on ImageNet. We report accuracy with 20-NN and 200-NN.

Method	20-NN (%)	200-NN (%)
NPID [26]	-	46.5
LA [31]	-	49.4
PCL [18]	54.5	-
VICReg [2]	64.5	62.8
SwAV [4] (with multi-crop)	65.7	62.7
Ours ($K = 4$)	**66.2**	**63.0**

Table 5. Evaluation of the representations pretrained with MSBReg on various downstream tasks: 1) the performance linear classifier on top of frozen ResNet-50 backbone and 2) object detection with fine-tuning. For the linear probing, we report mAP for VOC07 [10] benchmark, Top-1 accuracy (%) for Places [30] and iNaturalist2018 [24] benchmarks. For the object detection task, we report AP^{50}, AP^{75}, and AP^{all} for VOC07+12 benchmark.

Method	Classification (%)			VOC Detection		
	VOC07	Places	iNat18	AP^{50}	AP^{75}	AP^{all}
BoWNet [12]	79.3	51.1	-	81.3	61.1	53.5
MoCo v2 [15]	86.4	51.8	38.6	82.4	63.6	57.0
PIRL [19]	81.1	49.8	-	80.7	59.7	54.0
OBoW [13]	89.3	56.8	-	82.9	64.8	57.9
BYOL [14]	86.6	54.0	47.6	81.4	61.1	55.3
SimSiam [6]	-	-	-	82.4	63.7	57.0
Barlow Twins [29]	86.2	54.1	46.5	82.6	63.4	56.8
SwAV [4]	88.9	56.5	48.6	82.6	62.7	56.1
PixPro [28]	-	-	-	83.8	67.7	60.2
Ours ($K = 4$)	87.8	56.4	47.9	83.0	63.5	56.7

Table 6. Comparison of the quality of representations between BYOL [14] and ours on the STL-10 dataset [7]. The Top-1 classification accuracy is reported with different types of normalization techniques: a batch normalization (BN) [17] and a layer norm (LN) [1]. To see the effect of our proposed Brownian Diffusive Loss, $\mathcal{L}_b$, we also report scores of BYOL with $\mathcal{L}_b$ (4th row).

Method	Norm. Layer	Batch Size	λ_b	Top-1 (%)
BYOL	BN	256	0	89.5
Ours	BN	256	5×10^{-2}	91.4
BYOL	LN	256	0	10.6
BYOL + our $\mathcal{L}_b$	LN	256	5×10^{-3}	75.3
BYOL	LN	1024	0	10.6
Ours	LN	256	5×10^{-4}	80.7
Ours	LN	256	5×10^{-3}	82.3
Ours	LN	256	5×10^{-2}	78.7

For object detection task, we finetune pre-trained ResNet-50 backbone with the PASCAL VOC07+12 object detection benchmark [10]. We use Faster R-CNN [20] with C4 backbone as our baseline model. We report AP^{50}, AP^{75}, and AP^{all}. We observe in Table 5 that our model shows generally matched results compared with alternatives except for PixPro [28], which is proposed for . For AP_{50}, our method performs better than the baselines, while our method shows matched or slightly lower performance than other approaches.

We report instance segmentation result on COCO dataset in the appendix.

4.3. Brownian Diffusive Loss against Mode Collapse

BYOL [14] successfully uses only pairs of positives, but the reason why the online and target networks can avoid a so-called mode collapse, i.e. representations of all the examples are mapped to the same point in the embedding space, is not yet clearly explained. Existing work [11, 6, 22, 21] discussed that the use of the Batch Norm (BN) implicitly contributes to avoiding generating a collapsed representation. Especially, the original authors of [14] show that BYOL works without BN [21]. However, those methods are impractical in terms of restricting the network architecture design and this fact implies that these approaches are suboptimal. In our work, we propose to use Brownian diffusive loss, $\mathcal{L}_b$, which pushes embeddings into the radial direction to be uniformly sampled on the unit hyper-sphere. This helps to avoid collapsed representations without the need of using the Batch Norm (BN). We further discuss this in the appendix.

In Table 6, we empirically observe that BYOL suffers from a mode collapse when we replace the Batch Norm (in the prediction and projection heads) with another normalization technique, a Layer Norm (compare 1st vs. 3rd row). The top-1 classification accuracy is largely degraded from 89.5% to 10.6%, i.e. mode collapsed. Ours with the Brownian diffusive loss $\mathcal{L}_b$ was not the case (compare 2nd vs. 6th row). Though we observe a slight degradation in the top-1 classification accuracy, ours sufficiently avoid collapsed representations. Further, we evaluate the BYOL with our Brownian diffusive loss to demonstrate its effectiveness against a mode collapse. We observe that our Brownian diffusive loss helps avoid collapsed representations (compare 3rd vs. 4th rows). We also observe that the quality of representations depends on the strength of the hyper-parameter λ_b where we obtain the best performance with $\lambda_b = 5 \times 10^{-4}$. We observe a tension as we see a smaller or larger λ_b slightly degrades the quality of representations.

4.4. Comparison with Multi-Crop Method

We further compare Multiview centroid loss with the multi-crop method. The main difference between multiview centroid loss and multi-crop in SwAV is that our method uses the same resolution across all views while multi-crop uses low resolutions. We observe in Table 7 that a BYOL model with the multi-crop method shows a degradation (compare 1st vs. 2nd row), while MSBReg improves the performance of BYOL with a large margin (compare 1st vs. 3rd). This fact is also reported in [2].

Here, we describe the details of experiment. For a fair comparison, we implement the multi-crop method in the BYOL framework. The multi-crop method generates 2 views with full resolution (224×224 for ImageNet) and V views with low resolution (96×96 for ImageNet). Cropping small parts of an image is used to generate low-resolution images. We choose $V = 6$ following [4]. To apply the multi-crop method to BYOL, we reformulate BYOL loss (i.e. Eq. 1) as follows:

$$\begin{aligned}\mathcal{L}_{\text{mc-byol}}(\theta, \xi; \mathcal{X}) &:= \sum_{i,j}^{V+2} \left\| \hat{p}_i - \hat{z}_j{}' \right\|_2^2 \mathbb{1}(i \neq j) \\ &= \sum_{i,j}^{V+2} \left(2 - 2 \frac{\langle p_i, z'_j \rangle}{||p_i||_2 \cdot ||z'_j||_2} \right) \mathbb{1}(i \neq j)\end{aligned}$$

which shows that the multi-crop method minimizes the distance between pairs of embeddings, while our Multiview centroid loss minimizes the distance between each view and the geometric centroid of multi-views.

Table 7. Comparison accuracy of downstream image classification task on ImageNet between multiview loss and multi-crop [4]. We apply multi-crop method to BYOL.

Method	100 epochs	200 epochs	300 epochs
BYOL [14]	65.9	70.1	72.3
BYOL+multi-crop	65.8	68.7	70.3
Ours ($K = 4$)	70.2	73.6	74.4

Table 8. Ablation study to study the effect of our proposed three regularizations: (1) Multiview centroid loss $\mathcal{L}_c$, (2) Brownian diffusive loss $\mathcal{L}_b$, and (3) singular value loss $\mathcal{L}_w$. Note that we compare the top-1 classification accuracy (in %) of a linear classifier on the ImageNet-100 dataset.

Method	$\mathcal{L}_c$	$\mathcal{L}_b$	$\mathcal{L}_s$	Acc. (%)
BYOL	✗	✗	✗	71.92
BYOL	✗	✗	✓	72.84
BYOL	✗	✓	✗	72.84
BYOL	✗	✓	✓	72.41
Ours ($K = 4$)	✓	✗	✗	78.24
Ours ($K = 4$)	✓	✗	✓	79.68
Ours ($K = 4$)	✓	✓	✗	79.74
Ours ($K = 4$)	✓	✓	✓	80.38
Ours ($K = 8$)	✓	✗	✗	79.54
Ours ($K = 8$)	✓	✗	✓	79.96
Ours ($K = 8$)	✓	✓	✗	80.28
Ours ($K = 8$)	✓	✓	✓	81.56

4.5. Ablation Studies

Table 8 shows our ablation study to see the effect of our proposed three regularizations: (1) Multiview centroid loss $\mathcal{L}_c$, (2) Brownian diffusive loss $\mathcal{L}_b$, and (3) Singular value loss $\mathcal{L}_s$. Given the BYOL model as a baseline, we apply different combinations of our regularizations and measure the quality of representations following the linear evaluation protocol. We report scores on the ImageNet-100 dataset. We use ✓ and ✗ to indicate *with* and *without*, respectively. Note that we set λ_b and λ_w by default as 0.5 and 4.0×10^{-3}, respectively.

We first observe that a significant performance gain is obtained with our Multiview centroid loss $\mathcal{L}_c$ (compare 1st vs. 5th and 9th). The quality of the learned representations consistently improves as the number of views K increases. Since BYOL uses 2 views ($K = 2$) for training, doubling the number of views provides more than 6% performance gain. The other two regularizations, Brownian diffusive loss $\mathcal{L}_b$ and Singular value loss $\mathcal{L}_s$, also consistently improve the overall classification accuracy. For example, the classification performance improves 0.92% with the Brownian diffusive loss (compare 1st vs. 3rd) and the Singular value loss (compare 1st vs. 2nd). Such performance gain becomes more apparent with the Multiview centroid loss where we obtain a larger gain: 1.44% with the Singular value loss and 1.5% with the Brownian diffusive loss. Concretely, applying all our proposed regularizations together shows the best performance.

We further study the sensitivity of the tuning of the two new hyperparameters λ_s and λ_b. We report the result in the supplementary.

5. Conclusion

In this work, we have explored multiview, singular value regularization and Brownian diffusion methods for self-supervised learning. Each method implicitly induces contrastive effect, which stabilizes the the training of self-supervised learning. Our method achieves a good downstream task performance for instance classification as well as various transfer learning such as object detection, semantic segmentation.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

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

[5] 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.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings.

[8] 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.

[9] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. *CoRR*, abs/2007.06346, 2020.

[10] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[11] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with bootstrap your own latent (byol).

[12] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick P'erez, and Matthieu Cord. Learning representations by predicting bags of visual words. *2020 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 6926–6936, 2020.

[13] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Obow: Online bag-of-visual-words generation for self-supervised learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6826–6836, 2021.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[15] 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 (CVPR)*, June 2020.

[16] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[18] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[19] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[21] Pierre H. Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, and Michal Valko. Byol works even without batch statistics, 2020.

[22] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding, 2020.

[23] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks, 2021.

[24] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

[26] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[27] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. 2021.

[29] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021.

[30] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[31] Chengxu Zhuang, Alex Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6001–6011, 2019.