FroSSL: Frobenius Norm Minimization for Efficient Multiview Self-Supervised Learning

Oscar Skean[1], Aayush Dhakal[2],
Nathan Jacobs[2], and Luis Gonzalo Sanchez Giraldo[1]

[1] University of Kentucky
[2] Washington University in St. Louis
oscar.skean@uky.edu

Abstract. Self-supervised learning (SSL) is a popular paradigm for representation learning. Recent multiview methods can be classified as sample-contrastive, dimension-contrastive, or asymmetric network-based, with each family having its own approach to avoiding informational collapse. While these families converge to solutions of similar quality, it can be empirically shown that some methods are epoch-inefficient and require longer training to reach a target performance. Two main approaches to improving efficiency are covariance eigenvalue regularization and using more views. However, these two approaches are difficult to combine due to the computational complexity of computing eigenvalues. We present the objective function FroSSL which reconciles both approaches while avoiding eigendecomposition entirely. FroSSL works by minimizing covariance Frobenius norms to avoid collapse and minimizing mean-squared error to achieve augmentation invariance. We show that FroSSL reaches competitive accuracies more quickly than any other SSL method and provide theoretical and empirical support that this faster convergence is due to how FroSSL affects the eigenvalues of the embedding covariance matrices. We also show that FroSSL learns competitive representations on linear probe evaluation when used to train a ResNet18 on several datasets, including STL-10, Tiny Imagenet, and Imagenet-100. Github

1 Introduction

The problem of learning representations without human supervision is fundamental in machine learning. Unsupervised representation learning is particularly useful when label information is difficult to obtain or noisy. It requires the identification of structure in data with limited knowledge about what the structure is. One common way of learning structure without labels is joint embedding self-supervised learning (SSL) [4; 5; 14; 16; 17; 24; 33; 36]. The basic goal of SSL is to train neural networks to capture *semantic* input features that are *augmentation-invariant*. This goal is appealing for representation learning because the inference set often has similar semantic content to the training set.

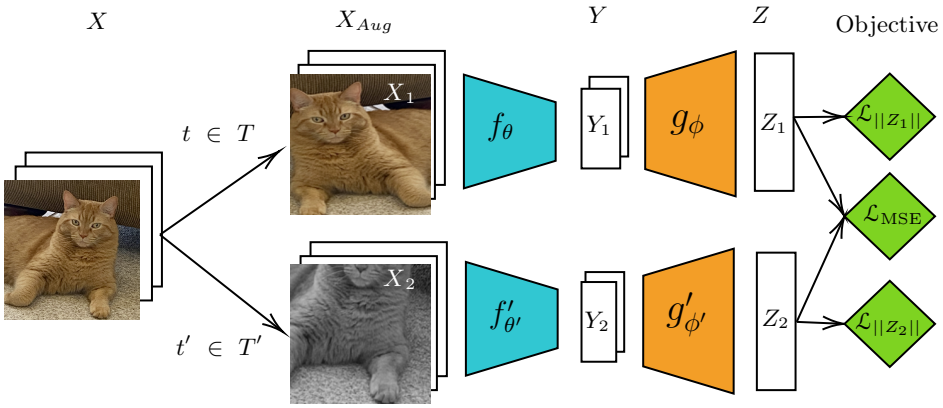

Fig. 1: The SSL pipeline used in this work. In general, the encoder and projector may be asymmetric. We use symmetric encoders with shared weights and the same augmentation set for each view. We refer to X_1 as view 1 of X, and X_2 as view 2. Only two views are shown here, though more may be used in practice.

A trivial solution to learning augmentation-invariant features is to encode all images to the same point, often called trivial or informational collapse. The resulting networks are essentially useless for downstream tasks. Different mechanisms have been proposed to handle collapse in SSL. These can be grouped into three families: sample-contrastive, dimension-contrastive, and asymmetric network methods.

A less studied problem in all current SSL methods is their speed of convergence. When compared to traditional supervised learning, SSL methods must be trained for large numbers of iterations to reach competitive performance on downstream tasks. For example, a typical experiment in the literature is to train for 1000 epochs on ImageNet which can take several weeks even with many GPUs. An imperative direction of research is to investigate how to reduce SSL training time. An observation that is often hidden by only reporting the final epoch accuracy is that, empirically, certain SSL methods require more training time to reach competitive accuracies. This phenomenon has been observed for many dimension-contrastive methods by Simon *et al.* [27] but not discussed in detail. We provide additional support for this claim in Section 4.1. Our work attempts to answer the following research question: Does there exist an SSL method with dimension-contrastive advantages, namely simplicity via avoidance of both negative sampling and architectural restrictions, while achieving competitive accuracies more quickly than other existing SSL methods?

We propose an SSL objective which we call FroSSL. Similar to many dimension-contrastive methods, FroSSL consists of a variance and invariance term. The invariance term is simply a mean-squared error between the views and is identical to VICReg's invariance term [1]. The variance term is the logarithm of the squared Frobenius norm of the normalized covariance embedding matrices. Using

the Frobenius norm of covariance matrices for improving learned representations has not been explored in SSL.

Our contribution can be summarized as:

- We introduce the FroSSL objective function and show that it is *both* dimension-contrastive and sample-contrastive up to a normalization of the embeddings.
- We introduce a theoretical framework that unifies dimension-contrastive methods that scale linearly in the number of views.
- We show that FroSSL combines two techniques to reduce training time: using more views and improving eigenvalue dynamics. We examine covariance eigenvalue trajectories during training on STL-10 to show that FroSSL learns useful, high-rank representations more quickly than other dimension-contrastive methods.
- We evaluate FroSSL on the standard setup of SSL pretraining and linear probe evaluation on CIFAR-10, CIFAR-100, STL-10, Tiny Imagenet, and Imagenet-100. We find that FroSSL achieves strong performance, especially when models are trained for fewer epochs.

2 Background and Notation

Consider a matrix $A \in \mathbb{R}^{m \times n}$. Let $A_{ij} \in \mathbb{R}$ denote the element at the ith row and jth column of A, and $A_{i,:} \in \mathbb{R}^m$ denote the ith column vector representing the ith row of A, and $A_{:,j}$ the jth column of A. Let $\sigma_k(A)$ denote the kth singular value of A ordered non-increasingly. The entry-wise product (also known as Hadamard product) is denoted as $A \odot B$. The Ky Fan p norm of A is defined as [18]:

$$\|A\|_p = \left(\sum_k^{\min(m,n)} \sigma_k^p(A) \right)^{1/p}, \tag{1}$$

which is a unitarily invariant norm. For $p = 2$, we have the Frobenius norm $\|A\|_2 = \|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$.

2.1 The Joint Embedding Self-Supervised Learning Problem

The goal of self-supervised learning is to learn useful representations without external supervision. Many visual joint embedding SSL methods follow a similar procedure which was first introduced in [4]. An example of this procedure is depicted in Figure 1. Let $\mathbf{X} = \{x_i\}_{i=1}^N$ be a mini-batch with N samples, V the number of augmented views, $T(\cdot)$ a function that applies randomly selected augmentations to an image, f a visual encoder network, and g a projector network.

First, each image $x_i \in \mathbf{X}$ is paired with augmented versions of itself, making the augmented dataset $\mathbf{X}_{\text{aug}} = \{T_1(x_i), \cdots, T_V(x_i)\}_{i=1}^N = \{X_1, \cdots, X_V\}$ With ideal augmentations, $(X_1)_{i,:}$ and $(X_2)_{i,:}$ have identical *semantic* content and different *style* content. Note that typically $V = 2$, but we make no such assumptions. For each augmentation, the embedding set is given by $Y_v = \{f((X_v)_{i,:})\}_{i=1}^N$

and projection set $Z_v = \{g((Y_v)_{i,:})\}_{i=1}^{N}$. Finally, an SSL objective is computed on the projections and backpropagated through both networks. The goal of the objective is to ensure that encoded augmentations for the same image are mapped close together by the projector, i.e. $(Z_a)_{i,:}$ and $(Z_b)_{i,:}$ are close in some sense of distance for all $a, b = 1, 2, \ldots, V$. At the same time, projections should capture the variability among images. Thus the goal of SSL is to train the networks f and g to extract *semantic* features that are invariant to any augmentations induced by $T(\cdot)$. In the following, we take a closer look at choices for the SSL objective.

2.2 The Three Families of Joint Embedding SSL Objectives

Objective functions for joint embedding self-supervised learning can be divided into three families. The first family consists of *sample-contrastive* methods [2; 4; 16; 17; 33] which use a contrastive loss to learn a representation that maps positive samples (augmentation of the same image) close together while pushing negative samples (different images) apart. These methods avoid collapse at the expense of making comparisons between positive and negative samples.

The second family consists of *asymmetric network* methods [3; 5; 14] which place restrictions on the architecture of the mapping network used, including asymmetrical encoders [5; 14], momentum encoders [17], and stop gradients [5; 15]. While these methods can achieve great results, they are rooted in implementation details and there is no clear theoretical understanding of how they avoid collapse [1].

The third, and most recent, family are the *dimension-contrastive* methods, which are sometimes called negative-free contrastive [32] or feature decorrelation methods [29]. These methods operate by reducing the redundancy in feature dimensions. Instead of examining *where* samples live in feature space, these methods examine *how* feature dimensions are being used. Methods in this family can avoid the use of negative samples while also not requiring restrictions in the network architecture to prevent collapse. Barlow Twins objective pushes the normalized cross-covariance between views towards the identity matrix [36]. VICReg consists of three terms: the invariance term enforces similarity in embeddings across views, while the variance/covariance terms regularize the covariance matrices of each view to prevent collapse [1]. W-MSE whitens and projects embeddings to the unit sphere before maximizing cosine similarity between positive samples [11]. I-VNE maximizes the von Neumann entropy of the embedding covariance matrices [20]. Finally, CorInfoMax maximizes the $\log\det$ entropy of both views while minimizing mean-squared error [25].

2.3 A Framework for Dimension-Contrastive Methods

Many recent works in dimension-contrastive SSL, whether explicitly or implicitly, consist of a combination of two competing objectives: an augmentation **invariance** term that pulls different augmentations from the same image close together, and a **variance** term that avoids collapse of the mapping by regulating

Table 1: Taxonomy of dimension-contrastive SSL methods describing how they avoid informational collapse and achieve augmentation invariance in the D_{inv} and D_{var} framework of Section 2.3.

Method	Variance $D_{\text{var}}(\Sigma_v \| \mathbf{I})$	Invariance $D_{\text{inv}}(Z_v, Z_r)$
VICReg	(Variance) Hinge loss on auto-covariance diagonal (Covariance) covariance off-diagonals per view $\sum_k^D \max\left(0, 1 - \sqrt{(\Sigma_v)_{k,k} + \epsilon}\right) + \nu \left\|\Sigma_v - \Sigma_v \odot \mathbf{I}\right\|_F^2$	MSE $\frac{1}{N}\left\|Z_v - Z_r\right\|_F^2$
W-MSE	Implicit through whitening that $D_{\text{var}}(\Sigma_v \| \mathbf{I}) = 0$ since $\Sigma_v = \mathbf{I}$ for all v	
CorInfoMax	Log Det Divergence: $D_{\log\det}(A \| B) = \text{trace}(AB^{-1}) - D - \log\det(AB^{-1})$ $D_{\log\det}(\Sigma_v + \epsilon\mathbf{I} \,\|\, \mathbf{I}) = \text{trace}(\Sigma_v + \epsilon\mathbf{I}) - D - \log\det(\Sigma_v + \epsilon\mathbf{I})$	
I-VNE	von Neumann Relative Entropy: $S_1(A \| B) = \text{trace}(A(\ln A - \ln B))$ $S_1(\Sigma_v \| \mathbf{I}) = \text{trace}(\Sigma_v \ln \Sigma_v)$	Cosine Similarity
FroSSL (ours)	2-Order Petz-Rényi Relative Entropy: $S_2(A \| B) = \log \text{trace}(A^2 B^{-1})$ $S_2(\Sigma_v \| \mathbf{I}) = \ln\left(\sum_{i=1}^N \sigma_i^2\right) = \ln \left\|\Sigma_v\right\|_F^2$	MSE $\frac{1}{N}\left\|Z_v - Z_r\right\|_F^2$

variance. Below, we unify dimension-contrastive methods into a general framework that is parameterized by choices of two distances. By carefully selecting these distances, specific dimension-contrastive methods can be recovered.

Let $Z_v \in \mathbb{R}^{N \times D}$ be a batch of projections and $\Sigma_v = \frac{1}{N}\hat{Z}_v^T \hat{Z}_v$ the corresponding covariance, where $\hat{Z}_v$ are the centered projections. A dimension-contrastive objective can be written as follows:

$$\min \sum_{v=1}^{V-1} \sum_{r=v+1}^{V} D_{\text{inv}}(Z_v, Z_r) + \gamma \sum_{v=1}^{V} D_{\text{var}}(\Sigma_v \| \mathbf{I}). \tag{2}$$

The first term of (2) is the **invariance** term which minimizes the distance $D_{\text{inv}} : \mathbb{R}^{N \times D} \times \mathbb{R}^{N \times D} \mapsto \mathbb{R}_{\geq 0}$ between all pairs of augmentations. The second term of (2) is a **variance** factor that forces the covariance of each augmentation to be close to identity according to a dissimilarity $D_{\text{var}} : \mathbb{R}^{D \times D} \times \mathbb{R}^{D \times D} \mapsto \mathbb{R}_{\geq 0}$. For instance, in VICReg [1], $D_{\text{inv}}(Z_v, Z_r) = \left\|Z_v - Z_r\right\|_F^2$ and $D_{\text{var}}(\Sigma_v \| \mathbf{I}) = \sum_k^D \max\left(0, 1 - \sqrt{(\Sigma_v)_{k,k} + \epsilon}\right) + \nu \left\|\Sigma_v - \Sigma_v \odot \mathbf{I}\right\|_F^2$. Similarly, in CorInfoMax [25] $D_{\text{inv}}(Z_v, Z_r)$ is the same as VICReg, but $D_{\text{var}}(\Sigma_v \| \mathbf{I})$ can be related to the log det divergence $D_{\log\det}(A \| B) = \text{trace}(AB^{-1}) - D - \log\det(AB^{-1})$ setting $A = \Sigma_v + \epsilon\mathbf{I}$ and B to a scaling of identity due to the normalization step in their projector. In Table 1, we show the dimension-contrastive methods which fit into this framework. We provide derivations in the Supp. Material.

Multiview Invariance Term In (2) the invariance term requires $V(V-1)/2$ comparisons which scales quadratically with the number of views. However, if $D_{\text{inv}}(Z_v, Z_r) = \left\|Z_v - Z_r\right\|_F^2$, then the invariance term may be simplified to

$$\sum_{v=1}^{V-1} \sum_{r=v+1}^{V} D_{\text{inv}}(Z_v, Z_r) = V \sum_{v=1}^{V} D_{\text{inv}}\left(Z_v, \overline{Z}\right) \tag{3}$$

Table 2: Taxonomy of dimension-contrastive SSL methods showing which desirable criteria they fulfill.

	Invariant to Projection Rotations	Manipulates Eigenvalues Explicitly	Quadratic in Batch Size and Dimension	Linear in Views
Barlow Twins	×	×	✓	×
VICReg	×	×	✓	✓
W-MSE	✓	×	×	✓
CorInfoMax	✓	✓	×	✓
I-VNE	✓	✓	×	✓
MMCR	✓	✓	×	✓
FroSSL (ours)	✓	✓	✓	✓

where $\overline{Z} = \frac{1}{V}\sum_{i=1}^{V} Z_i$ is the average projection across all views. If a method has a D_{inv} that can be rewritten this way, we say the method scales linearly with views. All methods displayed in Table 1 have this property.

3 The FroSSL Objective

To motivate FroSSL, we begin by posing four desirable criteria of dimension-contrastive methods.

1. **Invariant to Projection Rotations** We argue that dimension-contrastive methods should be invariant to rotations in the projections because the orientation of the covariance does not affect the relationships between principal components. In other words, redundancy in the embedding dimensions is invariant to the rotation of the embeddings. Thus the choices of D_{var} and D_{inv} should be rotationally invariant as well.
2. **Manipulates Eigenvalues Explicitly** Several works have shown that regularizing projection covariance eigenvalues in SSL can lead to reduced training time and improved downstream performance [15; 20; 34]. We provide empirical support for this in Section 4.1.
3. **Scales Quadratically in Batch Size and Dimension** The time complexity of the objective function scale *at most* quadratically with respect to N and D. This is often in opposition to the prior criteria which typically requires cubic eigendecomposition.
4. **Scales Linearly in Views** The time complexity of the objective function should be linear in the number of views V. This is advantageous because recent work has shown that using more views can reduce training time and improve downstream performance [2; 34]. We provide empirical support for this in Sections 4.2 and 5. Any dimension-contrastive method with D_{inv} that satisfies Equation (3) fulfills this criterion.

As shown in Table 2, no prior method meets all four criteria. We provide proofs in the Supp. Material. To construct a method that fulfills all criteria, we

modify the I-VNE objective function:

$$\max \mathcal{L}_{\text{I-VNE}} = \sum_{v=1}^{V} \operatorname{Tr} \Sigma_v \ln \Sigma_v + \sum_{v=1}^{V-1} \sum_{r=v+1}^{V} \frac{Z_v^T Z_r}{\|Z_v\|_2 \|Z_r\|_2} \tag{4}$$

The invariance term maximizes the cosine similarity between views. The variance term maximizes the von Neumann entropy of each view covariance matrix. The only criteria that I-VNE does not fulfill is being subcubic in batch size and dimension. This is due to the eigendecomposition needed to compute the matrix logarithm for the entropy. To begin addressing this, we first notice that the von Neumann entropy is a limit case of matrix-based α-order entropy [19; 26; 28]:

$$S_\alpha(\Sigma_v) = \frac{1}{1-\alpha} \log\left[\operatorname{Tr}(\Sigma_v^\alpha)\right] = \frac{1}{1-\alpha} \log\left(\sum_i^{\min(N,D)} \lambda_i^\alpha(\Sigma_v)\right) \tag{5}$$

Here, we do not require $\operatorname{trace}(\Sigma_v) = 1$ as is typically required by α-order entropy. The von Neumann entropy is equivalent to $S_1(\Sigma_v)$ in the limit. Another special case is collision entropy, given by $S_2(\Sigma_v)$ below:

$$S_2(\Sigma_v) = -\log\left(\sum_{i=1}^{\min(N,D)} \lambda_i^2(\Sigma_v)\right) = -\log(\|\Sigma_v\|_F^2) = -\log \sum_i \sum_j (\Sigma_v)_{ij}^2 \tag{6}$$

Notice how the left-hand side in the above equation explicitly uses the eigenvalues, while the right-hand side only uses matrix elements. This is made possible by the Frobenius norm, which offers an equivalency between a sum over eigenvalues and a sum over matrix elements. This has immediate impacts on the loss time complexity by relaxing the $O(\min(D,N)^3)$ eigendecomposition to the $O(\min(D,N)^2)$ Frobenius norm computation. The case of 2-order entropy is the only matrix-based α-entropy which does not require eigendecomposition. One potential downside is $S_2(\Sigma_v)$ does not penalize outlying eigenvalues as heavily as in $S_1(\Sigma_v)$. This is akin to the difference between mean-absolute error and mean-squared error. However, this has no significant impact in our experiments.

The variance term D_{var} for our objective will minimize the log Frobenius norm of normalized embeddings, causing the embeddings to spread out equally in all directions. Normalizing the embeddings is crucial because otherwise, minimizing the Frobenius norm will lead to trivial collapse. For the invariance term D_{inv}, we opt to use the mean-squared error between views. Our objective function FroSSL is given below:

$$\text{minimize } \mathcal{L}_{\text{FroSSL}} = \sum_{v=1}^{V} \log(\|\Sigma_v\|_F^2) + \gamma \left\|Z_v - \overline{Z}\right\|_F^2 \tag{7}$$

Note we simplify the pairwise mean-squared error via Equation (3). Because the Frobenius norm is invariant to transposition, we can choose to compute either $\left\|Z_v^T Z_v\right\|_F^2$ or $\left\|Z_v Z_v^T\right\|_F$ depending on if $D > N$. The former has time complexity $O(ND^2)$ while the latter has complexity $O(N^2 D)$. For consistency, we always use the former in our experiments. We provide pseudocode in the Supp. Material.

3.1 The Role of the Logarithm

The log in Equation (7) ensures that the contributions of the variance term to the gradient of the objective function become self-regulated ($\frac{d \log f(x)}{dx} = \frac{1}{f(x)} \frac{df(x)}{dx}$) with respect to the invariance term. We later compare the experimental performance of Equation (7) with and without the logarithms, showing that using logarithms leads to a gain in performance. Prior work has shown that Equation (7) with no logarithms causes dead neurons in the final encoder layer [20].

3.2 FroSSL is both Sample-contrastive and Dimension-contrastive

It can be shown, up to an embedding normalization, that FroSSL is both dimension-contrastive and sample-contrastive. First, we provide formal definitions of dimension-contrastive and sample-contrastive SSL, following Garrido *et al.* [13].

Definition 1 (Dimension-Contrastive Method). *An SSL method is said to be dimension-contrastive if it minimizes the non-contrastive criterion* $\mathcal{L}_{nc}(Z) = \left\|Z^T Z - diag(Z^T Z)\right\|_F^2$, *where* $Z \in \mathbb{R}^{N \text{x} D}$ *is a matrix of embeddings as defined above. This may be interpreted as penalizing the off-diagonal terms of the embedding covariance.*

Definition 2 (Sample-Contrastive Method). *An SSL method is said to be sample-contrastive if it minimizes the contrastive criterion* $\mathcal{L}_c(Z) = \left\|ZZ^T - diag(ZZ^T)\right\|_F^2$. *This may be interpreted as penalizing the similarity between pairs of different images.*

Next, we use the duality of the Frobenius norm, given by $\left\|Z^T Z\right\|_F = \left\|ZZ^T\right\|_F$, to show that FroSSL satisfies the qualifying criteria of both dimension-contrastive and sample-contrastive methods.

Proposition 1. *If every embedding dimension is normalized to have equal variance, then FroSSL is a dimension-contrastive method. See Supp. Material. for the proof.*

Proposition 2. *If every embedding is normalized to have equal norm, then FroSSL is a sample-contrastive method. See Supp. Material. for the proof.*

Proposition 3. *If the embedding matrices are doubly stochastic, then FroSSL is simultaneously dimension-contrastive and sample-contrastive.*

Proposition 3 allows for interpreting FroSSL as either a sample-contrastive or dimension-contrastive method, up to a normalization of the data embeddings. The choice of normalization strategy is not important to the performance of an SSL method [13]. Unless otherwise specified, we only normalize the variance and not the embeddings. These same proof techniques can be used to show that TiCo, MMCR, I-VNE, and CorInfoMax also belong to both families [20; 25; 34; 37]. Additionally, variants of the dimension-contrastive VICReg have been

proposed [13] that allow it to be rewritten as the sample-contrastive SimCLR. However, VICReg cannot be rewritten in such a way due to the hinge loss.

While Proposition 3 is interesting theoretically, it also offers empirical benefits to FroSSL. We examine overall wall-training time to reach competitive accuracies (4.3), robustness to augmentations (5.1), and performance on little pretraining data (5.2). In all of these experiments, sample-contrastive methods outperform dimension-contrastive methods. FroSSL shares the advantages observed empirically in sample-contrastive methods.

4 On Efficiency in Self-Supervised Learning

It is well-known that traditional SSL algorithms need hundreds or thousands of epochs to reach competitive accuracies. To compare the efficiency of different SSL algorithms, we can borrow theoretical and practical tools from the broader field of algorithmic complexity. In the context of machine learning, there are two measurements of particular interest to practitioners: *wall-time* needed to reach a given accuracy and VRAM *space* used. The former can be decomposed into the atomic quantities of average *wall-time per minibatch* and the *number of epochs* needed to reach a given accuracy. To emphasize why this decomposition matters, consider a scenario where wall-time per minibatch differs between two methods but overall wall-time does not. In such a scenario, using the method with the slower minibatch wall-time is advantageous for using fewer disk reads and less distributed network traffic. This is not obvious without observing the atomic quantities. Note we are careful to specify "epochs to reach a given accuracy" rather than "epochs to convergence". One reason for this is that classical experiments in SSL train for a fixed number of epochs rather than until convergence. Another reason is algorithms that more quickly reach a target performance, such as FroSSL or I-VNE, do not necessarily converge in fewer epochs.

The design of an SSL algorithm is a balancing act between minibatch time, space, and number of epochs. While conversations involving minibatch time and space have been rarely discussed in the SSL literature, discussion about the number of epochs has seen renewed interest [15; 27; 31]. However, if SSL algorithm design is indeed a balancing act of the three quantities above, then space and minibatch time deserve discussion too. Methods that boast reductions to one quantity may come with significant penalties to a different quantity. For example, dimension-contrastive methods use less space in practice than sample-contrastive methods, which prefer large minibatch sizes, or asymmetric methods like BYOL, which need an additional prediction network. However, the improved space efficiency comes at the cost of requiring a higher number of epochs [27]. In Sections 4.1 and 4.2, we discuss the advantages and drawbacks of two approaches to reducing the number of epochs. In Section 4.3, we compare a variety of SSL algorithms and visualize their time, space, and epoch tradeoffs.

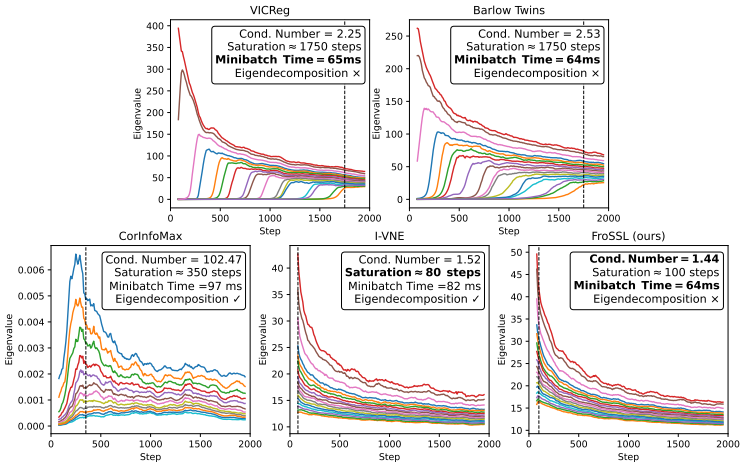

Fig. 2: The choice of variance term, $D_{\mathbf{var}}(\Sigma_v \| \mathbf{I})$, has a significant impact on training dynamics. Each subplot visualizes the trajectories of the top 20 eigenvalues of the embedding covariance matrix Σ_1 when trained with dimension-contrastive methods. These trajectories show how quickly Σ_v converges to γI, which has eigenvalues equal to $\frac{\gamma}{D}$. VICReg, Barlow Twins, and CorInfoMax converge slowly. FroSSL and I-VNE have similar training dynamics, but FroSSL has significantly lower computational complexity because it avoids explicitly computing the eigendecomposition.

4.1 Reducing the Number of Epochs with Eigenvalue Dynamics

Recent work has examined the training dynamics of SSL models [27]. In particular, they find that the eigenvalues of the covariance exhibit *stepwise* behavior, meaning that one eigendirection is learned at a time. This is readily seen in Figure 2 for VICReg and Barlow Twins. This phenomenon contributes to slowness in SSL optimization with the smallest eigendirections taking the longest to be learned. Other work shows that high-rank representations lead to better downstream performances [12]. It directly follows that if an SSL method requires a high number of epochs to learn high-rank representations, then it also needs a high number of epochs to learn useful representations.

We hypothesize that by carefully choosing the variance term $D_{\text{var}}(\Sigma_v \| \mathbf{I})$ to reduce stepwise eigenvalue dynamics, useful representations can be learned more quickly. Indeed, this behavior has already been observed in several SSL objectives already. CorInfoMax optimizes the log-determinant of each covariance Σ_v, which is defined as the log of the product of the Σ_v eigenvalues [25]. IsoLoss uses Σ_v eigenvalues as learning rate multipliers to equalize the convergence rate of different eigenmodes [15]. MMCR optimizes the nuclear norm of the average view embedding, which is defined as the sum of the singular value magnitudes [34]. I-VNE optimizes the von Neumann entropy of Σ_v, which is equal to the Shannon entropy of the Σ_v eigenvalues [20].

It is straightforward to show that FroSSL also directly influences the covariance eigenvalue dynamics. However, FroSSL is unique from prior methods because it does so while avoiding explicit eigendecomposition. This can be seen from Equation 6. Additionally, using the Frobenius norm eliminates numerical instabilities typically associated with eigendecomposition [9].

To highlight the existence and remedy of stepwise phenomena in practical scenarios, we create an experiment similar to the one used by Simon *et al*. [27]. In Figure 2, we plot the trajectories of the top 20 eigenvalues of Σ_1 when trained with different dimension-contrastive objectives. For all SSL objectives, a ResNet18 was trained for 5 epochs on STL-10 using SGD with $lr = 0.01$ and a batch size of 256. Further details are given in the Supp. Material.

The eigenvalues trajectories show how quickly Σ_v is approaching γI, which has eigenvalues equal to $\frac{\gamma}{D}$. We say that an objective is saturated once the stepwise learning phase is ended. This is marked as the step where λ_{20} has increased from zero and started decreasing. It is clear to see that SSL objectives that directly influence eigenvalues, namely CorInfoMax, I-VNE, and FroSSL, saturate much quicker than the others. Interestingly, the condition number for CorInfoMax, computed as $\frac{\lambda_1}{\lambda_{20}}$, is much larger than any other tested method. We hypothesize this is due to the choice of the ϵ hyperparameter for the regularization term when computing the determinant as $\det(\Sigma_1 + \epsilon I)$.

4.2 Reducing the Number of Epochs by Using More Views

Multiview with 3 or More Views In contrastive learning, using more views has been shown to have significant impacts on representation quality and downstream performance [30]. In SSL, using more augmentations for each image has the effect of averaging out noise from the mean embedding, which acts as a target for many SSL objectives as shown in Equation (3). This differs from increasing the batch size, which would instead average out noise across samples and not across targets. While using more views is promising, it has not seen widespread adoption in self-supervised learning. This is in part due to many sample-contrastive methods being quadratic in the number of views. However, this problem is circumvented for the dimension-contrastive methods shown in Table 1, which are instead linear in the number of views. One such method, W-MSE, has shown performance improvements when the number of views is increased from 2 to 4 [11]. Interestingly, MMCR is constant in the number of views because it operates only on the mean embedding [34].

Multi-Patch and Multi-Crop Methods An approach in a different vein is to extract and augment image patches to serve as views, rather than using full images. EMP-SSL has shown that this drastically reduces the number of epochs and overall wall-time needed to reach competitive accuracies by utilizing a bag-of-features model that embeds hundreds of small augmented patches per image [6; 31]. However, EMP-SSL comes at the cost of major penalities to time-per-minibatch and space in both training time and inference time.

Table 3: Comparison of the time/space/epoch tradeoffs for SSL algorithms trained on STL-10. FroSSL with 8 views achieves 80% top-1 accuracy in the least wall-time.

	SimCLR	MocoV2	BYOL	VICReg	Barlow	CorInfoMax	MMCR			FroSSL (ours)		
Num. Views	2	2	2	2	2	2	2	4	8	2	4	8
Loss Time Complexity	$O(V^2N^2)$			$O(VD^2)$	$O(V^2D^2)$	$O(VD^3)$	$O(\min(D,N)^3)$			$O(V\min(D,N)^2)$		
VRAM Space (GB)	**1.6**	2.8	2.0	**1.6**	**1.6**	1.7	1.7	2.9	5.3	1.7	2.9	5.3
Minibatch Wall-time (ms)	**60**	79	76	65	64	97	71	108	187	64	105	187
Number of Epochs to 80% Acc	347	180	187	360	370	405	380	211	63	290	144	**55**
Wall-time to 80% Acc (hours)	2.4	1.6	1.6	2.7	2.7	4.5	3.1	2.6	1.3	2.1	1.7	**1.2**

Table 4: Top-1 accuracies on STL-10 using an online linear classifier during training for specific numbers of epochs (left) and specific elapsed times (right).

	Epochs				
Method	3	10	30	50	100
SimCLR	40.7	44.8	61.5	66.2	70.1
MoCo v2	24.6	45.0	63.8	69.4	75.2
BYOL	28.8	32.7	59.6	64.7	70.6
VICReg	43.6	51.1	61.2	67.5	71.1
Barlow Twins	32.1	46.6	62.0	62.6	69.0
CorInfoMax	39.0	49.1	58.0	62.5	66.2
MMCR (2 views)	39.6	53.3	62.8	63.3	67.0
MMCR (4 views)	46.0	61.5	70.2	71.5	75.7
MMCR (8 views)	**51.1**	64.7	72.9	77.2	79.4
FroSSL (2 Views)	44.8	56.9	64.8	67.1	72.0
FroSSL (4 Views)	49.3	60.7	70.3	67.1	76.9
FroSSL (8 Views)	47.6	**65.5**	**74.5**	**78.4**	**81.8**

	Training Wall-Time (min.)		
Method	10	30	60
SimCLR	61.5	68.8	73.9
MoCo v2	57.4	70.0	76.2
BYOL	50.2	65.3	75.0
VICReg	63.4	70.3	72.9
Barlow Twins	55.5	66.1	68.0
CorInfoMax	56.1	64.9	65.6
MMCR (2 views)	54.7	68.4	70.9
MMCR (4 views)	**69.8**	**74.7**	76.6
MMCR (8 views)	64.7	73.5	78.0
FroSSL (2 Views)	63.4	69.7	74.9
FroSSL (4 Views)	**68.5**	73.7	76.3
FroSSL (8 Views)	58.6	74.1	**79.0**

As an alternative to using full-sized images or tiny patches for each view, multi-crop methods strike a balance [2]. A certain number of views are full-sized images while the remaining views are smaller crops. A typical setup for ImageNet is using two 224×224 views and six 96×96 views. These approaches differ from our experiments which use all full-sized views with FroSSL. However, we expect that FroSSL should work well as an objective function for these paradigms too.

4.3 Exploring Time, Space, and Epoch Tradeoffs

We now compare the efficiency of different SSL algorithms. We train a ResNet-18 for 500 epochs on STL-10 and measure the number of epochs needed to reach a top-1 accuracy of 80%. This threshold of 80% was chosen because all methods achieve that accuracy within 500 epochs. We used $N = 256$ and $D = 1024$ for all methods. Because these models were trained on a distributed cluster, it is important to compensate for different compute when measuring minibatch wall-time. In particular, we measure minibatch time by averaging over 2000 iterations of training on one NVIDIA A5000 GPU. We measure VRAM space as the maximum space requested by the training script. We calculate wall-time to 80% accuracy by multiplying minibatch time, epochs, and iterations per epoch.

In Table 3 we show the resources needed for each SSL objective. There are several observations to glean from this table. First, increasing the number of views reduces epochs and overall wall-time, even though space and minibatch time become larger. FroSSL with 8 views reaches 80% top-1 accuracy faster than any other tested method. Second, asymmetric methods require the least overall wall-time for any method using 2 views, at the cost of space. We hypothesize this

Table 5: Comparison of SSL methods on small datasets. All CorInfoMax and MMCR results are from our implementation. All Tiny ImageNet and STL-10 results are from our implementation. CIFAR-10 and CIFAR-100 results are reported from [8; 11]. IN-100 baseline results are from [8]. We observed negligible improvements from using more views for FroSSL on the CIFAR datasets; we used the 2-view CIFAR accuracies to compute 4/8 view averages. An asterisk (*) denotes Tiny ImageNet results where weak augmentations outperformed strong ones. Results within 0.5% of best are **bolded**.

Method	CIFAR-10	CIFAR-100	STL-10	Tiny-IN	IN-100	Average
Sample-Contrastive						
SimCLR	91.8	65.8	85.9	41.9	77.6	72.6
SwAV	89.2	64.9	82.6	41.2	74.3	70.5
MoCo v2	**92.9**	69.9	83.2	41.9	79.3	73.4
Asymmetric Network						
SimSiam	90.5	66.0	88.5	**45.6***	78.7	73.9
BYOL	**92.6**	**70.5**	88.7	40.1	**80.3**	74.4
DINO	89.5	66.8	78.9	34.9	78.9	69.8
Dimension-Contrastive						
VICReg	92.1	68.5	85.9	37.5	79.4	65.8
Barlow Twins	92.1	**70.9**	85.0	**45.3**	**80.2**	74.7
W-MSE 2	91.6	66.1	72.4	28.8*	69.0	65.6
CorInfoMax	**92.6**	69.7	83.1	43.9	74.7	72.8
I-VNE	89.7	65.7	87.4	**45.2**	77.6	73.1
MMCR (2 views)	88.6	65.8	84.3	41.2	76.7	71.3
MMCR (4 views)	89.6	67.3	88.2	42.8	78.8	73.3
MMCR (8 views)	89.3	68.3	90.3	43.2	**80.3**	74.2
FroSSL (2 views) [no log]	88.9	62.3	82.4	36.4	78.3	69.7
FroSSL (2 views)	**92.8**	**70.6**	87.3	44.2	78.2	74.6
FroSSL (4 views)	-	-	90.0	**45.3**	79.4	75.6
FroSSL (8 views)	-	-	**90.9**	**45.3**	**79.8**	**75.9**

is due to enhanced training stability from momentum encoders. Third, doubling the number of views does not necessarily double the minibatch wall-time. This is because some parts of the training script, such as data loading and logging, do not get slower as the number of views increases. In Table 4, we show top-1 accuracies over epochs and over time. In both scenarios, FroSSL with 8 views has the highest accuracy after training is finished.

5 Experimental Results

In this section, we use a linear probe to evaluate learned representations on CIFAR-10 [22], CIFAR-100, STL-10 [7], Tiny ImageNet [23], and ImageNet-100 [30]. Our implementation is based on the solo-learn SSL framework [8].

In Table 5, we show linear probe evaluation results on these datasets. It is readily seen that FroSSL learns competitive representations in comparison to other SSL methods. The implementation details can be summarized as:

- **Optimizer** The backbone uses LARS optimizer [35] with an initial learning rate of 0.3, weight decay of 1e-6, and a warmup cosine learning rate scheduler. The linear probe uses the SGD optimizer [21] with an initial learning rate of 0.3, no weight decay, and a step learning rate scheduler with decreases at 60 and 80 epochs.
- **Epochs** For CIFAR-10 and CIFAR-100, we pretrain the backbone for 1000 epochs. For STL-10, we pretrain for 500 epochs. For Tiny Imagenet, we pretrain for 800 epochs. For Imagenet-100, we pretrain for 800 epochs. All linear probes were trained for 100 epochs.

Table 6: The top-1% accuracies after training on Tiny-Imagenet using weak or strong augmentations.

Method	Weak	Strong	Δ%
SimCLR	39.5	41.9	2.4
SwAV	39.9	41.2	1.5
MoCo v2	40.9	41.9	1.0
SimSiam	45.6	39.7	-5.9
BYOL	39.4	40.1	0.7
DINO	32.2	34.9	2.7
VICReg	18.1	37.5	19.6
Barlow Twins	36.8	45.3	8.5
CorInfoMax	33.1	43.9	10.8
MMCR (2 views)	24.2	41.2	17.0
FroSSL (2 views)	39.4	44.2	4.8

Table 7: Accuracies after pretraining on ImageNet-1k for 100 epochs with only 10% of data.

	Top-1	Top-5
SimCLR	31.1	56.6
BYOL	12.7	29.1
SimSiam	22.7	46.3
Barlow Twins	23.6	46.9
FroSSL (2 Views)	33.4	59.1
FroSSL (8 Views)	38.2	64.1

– **Hyperparameters** A batch size of N=256 is used for all datasets, except for Tiny ImageNet which used N=512. For FroSSL, we used $\gamma = 1.4$ for 2 views and $\gamma = 2$ for 4 and 8 views. We used an MLP with output dimension $D = 1024$ for FroSSL. Details about augmentations and method-specific hyperparameters are given in the Supp. Material.

5.1 Robustness to Augmentations

We trained models on Tiny ImageNet using both weak and strong augmentations. Weak augmentations had Gaussian blur probabilities (0.5, 0.5) and solarization probabilities (0, 0) for each view. Strong augmentations had Gaussian blur probabilities (1.0, 0.1) and solarization probabilities (0.2, 0). As shown in Table 6, FroSSL is more robust to changes in augmentations than any other dimension-contrastive method.

5.2 Performance In Low Data Regime

We trained models on ImageNet-1K [10] using only 10% of the data and evaluated them using the standard linear probe. Note that limited data was used in both pretaining and evaluation. As shown in Table 7, FroSSL achieves a better downstream performance on limited data than any other tested method.

6 Conclusion

We introduced FroSSL, a self-supervised learning method that can be seen as both sample- and dimension-contrastive. We showed that FroSSL enjoys the simplicity of dimension-contrastive methods while achieving the empirical advantages of sample-contrastive methods. In particular, we discovered that FroSSL can achieve substantially stronger performance than alternative SSL methods when trained with less overall wall-time. To better understand why this is happening, we presented empirical results based on eigenvalue trajectories. We demonstrated the effectiveness of FroSSL through extensive experiments on standard datasets.

Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2021-2011000005 and the Office of the Under Secretary of Defense for Research and Engineering under award number FA9550-21-1-0227. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the U.S. Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Bibliography

[1] Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-invariance-covariance regularization for self-supervised learning. In: International Conference on Learning Representations (2022)

[2] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems **33** (2020)

[3] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)

[4] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)

[5] Chen, X., He, K.: Exploring simple Siamese representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)

[6] Chen, Y., Bardes, A., LI, Z., LeCun, Y.: Bag of image patch embedding behind the success of self-supervised learning. Transactions on Machine Learning Research (2023)

[7] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics. pp. 215–223 (2011)

[8] da Costa, V.G.T., Fini, E., Nabi, M., Sebe, N., Ricci, E.: solo-learn: A library of self-supervised methods for visual representation learning. Journal of Machine Learning Research **23**(56), 1–6 (2022)

[9] Dang, Z., Yi, K.M., Hu, Y., Wang, F., Fua, P., Salzmann, M.: Eigendecomposition-free training of deep networks with zero eigenvalue-based losses. In: European Conference on Computer Vision. pp. 768–783 (2018)

[10] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)

[11] Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International Conference on Machine Learning. pp. 3015–3024 (2021)

[12] Garrido, Q., Balestriero, R., Najman, L., Lecun, Y.: RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In: International Conference on Machine Learning. pp. 10929–10974 (2023)

[13] Garrido, Q., Chen, Y., Bardes, A., Najman, L., LeCun, Y.: On the duality between contrastive and non-contrastive self-supervised learning. In: International Conference on Learning Representations (2023)

[14] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33** (2020)
[15] Halvagal, M.S., Laborieux, A., Zenke, F.: Implicit variance regularization in non-contrastive ssl. Advances in Neural Information Processing Systems **36** (2023)
[16] HaoChen, J.Z., Wei, C., Gaidon, A., Ma, T.: Provable guarantees for self-supervised deep learning with spectral contrastive loss. Advances in Neural Information Processing Systems **34** (2021)
[17] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
[18] Horn, R.A., Johnson, C.R.: Matrix Analysis: Second Edition. Cambridge university press (2013)
[19] Hoyos-Osorio, J.K., Sanchez-Giraldo, L.G.: The representation Jensen-Shannon divergence. arXiv preprint arXiv:2305.16446 (2023)
[20] Kim, J., Kang, S., Hwang, D., Shin, J., Rhee, W.: VNE: An effective method for improving deep representation by manipulating eigenvalue distribution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3799–3810 (2023)
[21] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
[22] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
[23] Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
[24] Li, Y., Pogodin, R., Sutherland, D.J., Gretton, A.: Self-supervised learning with kernel dependence maximization. Advances in Neural Information Processing Systems **34** (2021)
[25] Ozsoy, S., Hamdan, S., Arik, S., Yuret, D., Erdogan, A.: Self-supervised learning with an information maximization criterion. Advances in Neural Information Processing Systems **35** (2022)
[26] Sanchez Giraldo, L.G., Rao, M., Principe, J.C.: Measures of entropy from data using infinitely divisible kernels. IEEE Transactions on Information Theory **61**(1), 535–548 (2015)
[27] Simon, J.B., Knutins, M., Liu, Z., Geisz, D., Fetterman, A.J., Albrecht, J.: On the stepwise nature of self-supervised learning. In: International Conference on Machine Learning (2023)
[28] Skean, O., Osorio, J.K.H., Brockmeier, A.J., Giraldo, L.G.S.: DiME: Maximizing mutual information by a difference of matrix-based entropies. arXiv preprint arXiv:2301.08164 (2023)
[29] Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., Dai, J.: Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14431–14440 (2022)

[30] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: European Conference on Computer Vision. pp. 776–794 (2020)
[31] Tong, S., Chen, Y., Ma, Y., Lecun, Y.: EMP-SSL: Towards self-supervised learning in one training epoch. arXiv preprint arXiv:2304.03977 (2023)
[32] Tsai, Y.H.H., Bai, S., Morency, L.P., Salakhutdinov, R.: A note on connecting barlow twins with negative-sample-free contrastive learning. arXiv preprint arXiv:2104.13712 (2021)
[33] Tsai, Y.H.H., Wu, Y., Salakhutdinov, R., Morency, L.P.: Self-supervised learning from a multi-view perspective. In: International Conference on Learning Representations (2021)
[34] Yerxa, T., Kuang, Y., Simoncelli, E., Chung, S.: Learning efficient coding of natural images with maximum manifold capacity representations. Advances in Neural Information Processing Systems **36** (2024)
[35] You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
[36] Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320 (2021)
[37] Zhu, J., Moraes, R.M., Karakulak, S., Sobol, V., Canziani, A., LeCun, Y.: TiCo: Transformation invariance and covariance contrast for self-supervised visual representation learning. arXiv preprint arXiv:2206.10698 (2022)