FroSSL: Frobenius Norm Minimization for Self-Supervised Learning

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

Abstract

Self-supervised learning (SSL) is an increasingly popular paradigm for representation learning. Recent 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 dimension-contrastive methods converge to similar solutions as sample-contrastive methods, it can be empirically shown that some methods require more epochs of training to converge. Motivated by closing this divide, we present the objective function FroSSL which is both sample- and dimension-contrastive up to embedding normalization. FroSSL works by minimizing covariance Frobenius norms for avoiding collapse and minimizing mean-squared error for augmentation invariance. We show that FroSSL converges more quickly than a variety of other SSL methods 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 the CIFAR-10, CIFAR-100, STL-10, and ImageNet datasets.

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 without any preconceptions about what the structure is. One way of learning structure without labels is self-supervised learning (SSL). A flurry of SSL approaches have been proposed for learning visual representations (Chen et al., 2020; HaoChen et al., 2021; Tsai et al., 2021b; Chen & He, 2021; Grill et al., 2020; He et al., 2020; Zbontar et al., 2021; Li et al., 2021). 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.

A trivial solution to learning augmentation-invariant features is to learn networks that encode *every* image to the same point. Such a solution is known as informational collapse and is of course useless for downstream tasks. SSL approaches can be roughly divided into three families, each with its own method of avoiding collapse: sample-contrastive (SC) methods, dimension-contrastive (DC) methods, and asymmetric network (AN) methods. One disadvantage common to 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 convergence. For example, a typical experiment in the literature is to train for 1000 epochs on ImageNet which can take several weeks even with 4 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 seem to converge slower than others. This phenomenon has been

[1] University of Kentucky [2] University of Delaware
Correspondence to oscar.skean@uky.edu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Table 1: Taxonomy of dimension-contrastive SSL methods describing how they avoid informational collapse and achieve augmentation invariance

Method	Variance	Invariance
Barlow Twins	Cross-correlation off-diagonals	Cross-correlation diagonals
VICReg	(Variance) Hinge loss on auto-covariance diagonal (Covariance) covariance off-diagonals per view	MSE
W-MSE	Implicit through whitening	MSE
CorInfoMax	Log-determinant entropy of covariance per view	MSE
FroSSL (ours)	Log of normalized covariance Frobenius norm per view	MSE

observed in Simon et al. (2023) but not discussed in detail. Our goal is 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 a superior speed of convergence to other existing SSL methods?

We propose an SSL objective which we call FroSSL. Similar to many DC 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 (Bardes et al., 2022). The variance term is the log of the squared Frobenius norm of the normalized covariance embedding matrices. To the best of our knowledge, using the Frobenius norm of covariance matrices 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 evaluate FroSSL on the standard setup of SSL pretraining and linear probe evaluation on CIFAR-10, CIFAR-100, STL-10, and ImageNet. We find that FroSSL achieves strong performance, especially when models are trained for fewer epochs.
- We examine the covariance eigenvalues of various SSL methods to show which methods lead to the best-conditioned, and thus quickest, optimization problems.

2 Dimension-Contrastive Methods

The dimension-contrastive methods, which are sometimes called negative-free contrastive (Tsai et al., 2021a) or feature decorrelation methods (Tao et al., 2022), 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. Many recent works in DC SSL, whether explicitly or implicitly, consist of having a loss function that fulfills two roles:

- **Variance** This is the *explosion factor* that ensures informational collapse is avoided.
- **Invariance** This is the *implosion factor* that ensures useful augmentation-invariant representations are learned.

SSL methods belonging to the DC family include Barlow Twins (Zbontar et al., 2021), VICReg (Bardes et al., 2022), W-MSE (Ermolov et al., 2021), and CorInfoMax (Ozsoy et al., 2022). A taxonomy of these methods showing how they avoid informational collapse and achieve augmentation invariance is shown in Table 1.

3 The FroSSL Objective

To motivate FroSSL, we begin by examining the Barlow Twins objective,

$$\mathcal{L}_{\text{Barlow}} = \sum_i (1 - M_{ii})^2 + \lambda \sum_i \sum_{i \neq j} {M_{ij}}^2 \tag{1}$$

where M is the cross-correlation matrix. Without feature normalization, the objective $\mathcal{L}_{\text{Barlow}}$ pushes M to approach identity and is not rotationally invariant. However, we posit that DC methods *should* be rotationally invariant 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 DC methods should be rotationally invariant as well.

One natural matrix operation that is invariant to unitary transformations is the Frobenius norm. Minimizing the Frobenius norm of normalized embeddings will cause the embeddings to spread out equally in all directions. Normalization is crucial because otherwise, minimizing the Frobenius norm will lead to trivial collapse. We propose to use the following term to reduce redundancy between dimensions:

$$\mathcal{L}_{\text{Fro}} = \log(||Z_1^T Z_1||_F^2) + \log(||Z_2^T Z_2||_F^2) \tag{2}$$

where Z_i is a batch of embeddings in view i. The Frobenius norm $||\cdot||_F$ is defined as:

$$||A||_F = \sqrt{\sum_i^m \sum_j^n A_{ij}^2} = \sqrt{\sum_k^{\min(m,n)} \sigma_k^2(A)} \tag{3}$$

where $\sigma_k(A)$ is the kth largest singular value of A. The $\mathcal{L}_{\text{Fro}}$ term fills the role of a variance term. For the invariance term, we can simply use mean-squared error between the views. Combining the variance and invariance terms yields the FroSSL objective.

$$\text{minimize } \mathcal{L}_{\text{FroSSL}} = \log(||Z_1^T Z_1||_F^2) + \log(||Z_2^T Z_2||_F^2) + \frac{1}{N}\sum_{i=1}^n ||z_{1,i} - z_{2,i}||_2^2 \tag{4}$$

By the duality of the Frobenius norm, we can choose to calculate either $||Z_1^T Z_1||_F^2$ or $||Z_1 Z_1^T||_F$ depending on if $d > n$. The former has time complexity $O(nd^2)$ while the latter has complexity $O(n^2 d)$. We provide Pytorch-style pseudocode in Appendix A.

3.1 The Role of the Logarithm - Entropy and Self Regulation

The role of the logarithms in (2) is twofold. First, the logarithm allows interpreting $\mathcal{L}_{\text{Fro}}$ as entropy maximization. One recent information-theoretic framework with success in deep learning is matrix-based entropy (Sanchez Giraldo et al., 2015). It is an information-theoretic quantity that behaves similarly to Rényi's α-order entropy, but it can be estimated directly from data without making strong assumptions about the underlying distribution. In particular, the first and second terms of (2) correspond to the matrix-based negative collision entropies of Z_1 and Z_2. This is relevant because maximizing collision entropy directly minimizes the coincidence of points, thus avoiding trivial collapse. We compare FroSSL to other methods using entropy in Appendix C.

Second, we hypothesize that the $\log$ 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. Initially we attempted using tradeoffs between the Frobenius norm term and the mean-squared error term. However, a grid search showed that the optimal tradeoff was when the terms were equally weighted. This is a nice advantage over methods such as Barlow Twins and VICReg, where the choice of tradeoff hyperparameters is crucial to the performance of the model.

3.2 FroSSL is both Sample-contrastive and Dimension-contrastive

Using the definitions of SC and DC methods first proposed in Garrido et al. (2023b), it can be shown that, up to an embedding normalization, FroSSL is both SC and DC. Full proofs of this claim are given in Appendix F.1. The choice of normalization strategy is not of particular importance to the performance of an SSL method (Garrido et al., 2023b). Unless otherwise specified, we only normalize the variance and not the embeddings. Another method that shares these properties is TiCo (Zhu et al., 2022). Additionally, variants of the DC VICReg were introduced in Garrido et al. (2023b) that allowed it to be rewritten as the SC SimCLR.

4 Comparing Stepwise Convergence in the Nonlinear Regime

Recent work has examined the training dynamics of SSL models (Simon et al., 2023). In particular, they find that the eigenvalues of the covariance exhibit "stepwise" behavior, meaning that one eigendirection is learned at a time. They claim that this phenomenon contributes to slowness in SSL

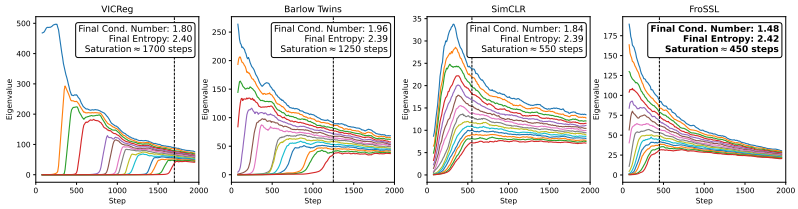

Figure 1: The top 14 eigenvalues of the embedding covariance $Z_1^T Z_1$. The condition number and eigenvalue Shannon entropy are shown for the end of epoch 5 (roughly 2000 steps). A vertical line marks the saturation of the 14^{th} eigenvalue. The best quantities are **bolded**.

Table 2: **(Left)** The top1 accuracies of SSL methods on small datasets. **(Right)** top1 accuracies at certain epochs on STL-10 using an online linear classifier during training. Best result is in **bold**, second and third best are underlined.

Method	CIFAR-10	CIFAR-100	STL-10	Average
Sample-Contrastive				
SimCLR	91.8	65.8	85.9	81.2
SwAV	89.2	64.9	82.6	78.9
MoCo v2	**92.9**	69.9	83.2	82.0
Asymmetric Network				
SimSiam	90.5	66.0	88.5	81.7
BYOL	92.6	70.5	**88.7**	**83.9**
DINO	89.5	66.8	78.9	78.4
Dimension-Contrastive				
VICReg	92.1	68.5	85.9	82.2
Barlow Twins	92.1	**70.9**	85.0	82.7
W-MSE 2	91.6	66.1	72.4	76.7
FroSSL	92.8	70.6	87.3	83.6

3	10	30	50	100
40.7	44.8	61.5	66.2	70.1
30.9	38.7	64.6	69.3	74.3
24.6	45.0	63.8	**69.4**	75.2
31.8	41.2	54.7	65.6	**77.1**
28.8	32.7	59.6	64.7	70.6
26.6	26.7	38.2	43.2	46.1
43.6	51.1	61.2	67.5	71.1
32.1	46.6	62.0	62.6	69.0
17.2	30.4	45.6	53.4	61.9
44.8	**56.9**	**64.8**	67.1	72.0

optimization because the smallest eigendirections take longer to be learned. This is supported by a recent finding that shows that high-rank representations lead to better classification accuracies (Garrido et al., 2023a).

We create an experimental setup similar to the one used in Simon et al. (2023). In Figure 4, we compare FroSSL to VICReg, Barlow Twins, and SimCLR. We train for 5 epochs and plot the top 14 eigenvalues of the view 1 covariance $Z_1^T Z_1$ over time. At the end of the 5th epoch, FroSSL outperforms the other methods in the following three metrics: condition number, eigenvalue entropy, and saturation time. We speculate that FroSSL allows the covariance eigenvalues to converge quicker because per Equation (3), the $\mathcal{L}_{\text{Fro}}$ term can be rewritten in terms of the embedding matrix eigenvalues. This shows that if each dimension is normalized to have variance ρ, then $\mathcal{L}_{\text{Fro}}$ explicitly tries to make the covariance eigenvalues approach to ρ. Implementation and metric details are given in D.1.

5 Experimental Results

For CIFAR-10, CIFAR-100, and STL-10, we use the solo-learn framework (da Costa et al., 2022). For methods other than FroSSL, we show CIFAR-10 and CIFAR-100 results from da Costa et al. (2022); Ermolov et al. (2021). All STL-10 models and results were trained by us. Implementation details are in Appendix *D.2*. In the left of Table 2, we show linear probe evaluation results on these datasets. It is readily seen that FroSSL learns competitive representations with other SSL methods. In the right of Table 2, online linear classifier accuracies are shown for STL-10 on several epochs during training. FroSSL outperforms all other DC methods. Additionally, FroSSL outperforms *all* other SSL methods shown in the first 30 epochs. In Appendix B, we show a comparison between FroSSL and Barlow Twins on training a ResNet18 on ImageNet.

6 Conclusion

We introduced FroSSL, a self-supervised learning method that can be seen as both sample- and dimension-contrastive. We demonstrated its effectiveness through extensive experiments on standard datasets. In particular, we discovered that FroSSL is able to achieve substantially stronger performance than alternative SSL methods when trained for a small number of epochs. To better understand why this is happening, we presented empirical results based on stepwise eigendecompositions. We plan to examine the theoretical convergence properties of FroSSl, similarly to the analysis done in Simon et al. (2023). We will also try FroSSL in combination with other SSL methods as a way of achieving faster convergence.

References

Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*, 35:26671–26685, 2022. 10

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. 2, 8

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020. 1, 9

Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 1

Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6, 2022. URL http://jmlr.org/papers/v23/21-1155.html. 4, 8

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pp. 3015–3024. PMLR, 2021. 2, 4

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, pp. 10929–10974. PMLR, 2023a. 4

Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann LeCun. On the duality between contrastive and non-contrastive self-supervised learning. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=kDEL91Dufpa. 3, 11

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021. 1

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020. 1

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

Lyudmyla F Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987. 8

Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34: 15543–15556, 2021. 1

Serdar Ozsoy, Shadi Hamdan, Sercan Arik, Deniz Yuret, and Alper Erdogan. Self-supervised learning with an information maximization criterion. *Advances in Neural Information Processing Systems*, 35:35240–35253, 2022. 2, 8

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008. 12

Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C Principe. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1):535–548, 2015. 3

James B. Simon, Maksis Knutins, Ziyin Liu, Daniel Geisz, Abraham J. Fetterman, and Joshua Albrecht. On the stepwise nature of self-supervised learning. In *International Conference on Machine Learning*, 2023. 2, 3, 4, 10, 11, 12

Chenxin Tao, Honghui Wang, Xizhou Zhu, Jiahua Dong, Shiji Song, Gao Huang, and Jifeng Dai. 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. 2

Yao-Hung Hubert Tsai, Shaojie Bai, Louis-Philippe Morency, and Ruslan Salakhutdinov. A note on connecting barlow twins with negative-sample-free contrastive learning. *arXiv preprint arXiv:2104.13712*, 2021a. 2

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=-bdp_8Itjwp. 1

Zhengyu Yang, Zijian Hu, Xuefeng Hu, and Ram Nevatia. SimMER: Simple maximization of entropy and rank for self-supervised representation learning, 2022. URL https://openreview.net/forum?id=77_zstKV8HQ. 8

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 8

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021. 1, 2, 8, 9

Jiachen Zhu, Rafael M Moraes, Serkan Karakulak, Vlad Sobol, Alfredo Canziani, and Yann LeCun. Tico: Transformation invariance and covariance contrast for self-supervised visual representation learning. *arXiv preprint arXiv:2206.10698*, 2022. 3

A Pseudocode for FroSSL

```
for x in loader:
    # augment the image
    x_a, x_b = augment(x)

    # pass through network f to get embeddings
    z_a = f(x_a)
    z_b = f(x_b)
    N, d = Z_a.shape

    # center embeddings
    Z_a = Z_a - Z_a.mean(0)
    Z_b = Z_b - Z_b.mean(0)

    # normalize dimensions to sqrt(D) std.
    Z_a = (D**0.5) * (Z_a / Z_a.norm())
    Z_b = (D**0.5) * (Z_b / Z_b.norm())

    # calculate invariance (MSE) term
    invariance_loss = MSELoss(Z_a, Z_b)

    # calculate variance (Frobenius norm) term
    frobenius_a = torch.log(torch.norm(Z_a.T @ Z_a, ord='fro'))
    frobenius_b = torch.log(torch.norm(Z_b.T @ Z_b, ord='fro'))
    variance_loss = frobenius_a + frobenius_b

    # FroSSL loss
    loss = invariance_loss + variance_loss
    loss.backward()
    optimizer.step()
```

B Performance on ImageNet

Here we use FroSSL to train a ResNet18 on ImageNet for 100 epochs. We compare to Barlow Twins on the exact same setup. We show the top1 and top5 accuracies in the first 30 epochs in Figure B. Even after the first epoch, FroSSL has an improvement of 12.2% over Barlow Twins. We show the first 30 epochs to emphasize what happens early in training. Afterward, Barlow Twins does catch up to FroSSL and achieves similar performances. FroSSL and Barlow Twins achieve final top1/top5 accuracies of $53.4/77.7$ and $52.5/77.5$, respectively. The important implementation details are given in Appendix $D.3$.

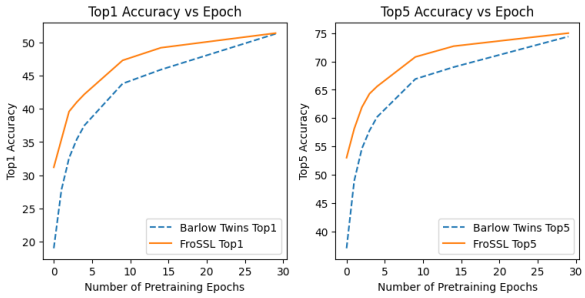

Figure 2: Comparison of SSL methods when training a ResNet18 on ImageNet.

C Entropy in SSL

Here we briefly discuss other SSL methods that, like FroSSL, can be interpreted as using entropy. The FroSSL objective is closely related to the CorInfoMax objective proposed in Ozsoy et al. (2022). The CorInfoMax objective uses $\log\det$ entropy, as opposed to the matrix-based entropy described in Section 3.1. One advantage of our approach is that the Frobenius norm can be computed in $O(d^2n)$, assuming that $d < n$. On the other hand, $\log\det$ entropy always requires computing the eigendecomposition which is $O(d^3)$. Another advantage of FroSSL over CorInfoMax is the absence of hyperparameters in the objective. We found the selection of ϵ to be critical for avoiding instabilities in the eigendecomposition.

Another recent SSL work that uses entropy is SimMER (Yang et al., 2022). Rather than $\log\det$ or matrix-based entropy, SimMER uses an entropy estimator based on nearest neighbors (Kozachenko & Leonenko, 1987). Unlike FroSSL, SimMER is not negative-free because the estimator implicitly chooses the nearest neighboring point as a negative.

D Experimental Details

D.1 Stepwise Convergence Experimental Details

For all SSL objectives, a ResNet18 was trained on STL10 using $lr = 0.1$ and a batch size of 256. The learning rate was chosen by performing a sweep over {1e-4, 1e-3, 1e-2, 1e-1} and selecting the one that led to the highest linear probe accuracy after 100 epochs. A learning rate of 0.1 was best for all objectives. Training occurred for only 5 epochs because we were interested in stepwise behaviors early during training.

The hyperparameters for each SSL criterion are:

- **Barlow Twins** We used $\lambda = 0.05$ as recommended by Zbontar et al. (2021) and $d = 1024$.
- **VICReg** We used $\lambda = 25$, $\mu = 25$, $\nu = 1$ as recommended by Bardes et al. (2022).
- **SimCLR** We used temperature $\tau = 0.2$ and $d = 256$.
- **FroSSL** We used $d = 1024$.

The metrics we use to compare methods are:

- **Condition Number** Given by $\frac{\lambda_1(Z_1^T Z_1)}{\lambda_{14}(Z_1^T Z_1)}$. The ideal condition number is 1 because the smallest eigendirection is as relevant as the largest.
- **Shannon Entropy** Given by $-\sum_i \lambda_i \log(\lambda_i)$, where the eigenvalues are normalized to sum to 1 before computation. The optimal value here is maximum entropy, which is obtained when all eigenvalues are equal. Higher entropy is better because more eigendirections have been learned.
- **Saturation** Given by the step at which the 14th eigenvalue saturates. Earlier is better because convergence can occur with fewer training steps.

D.2 Small Datasets Experimental Details

Optimizer For the backbone, the LARS optimizer (You et al., 2017) is used with an initial learning rate of 0.3, weight decay of 1e-6, and a warmup cosine learning rate scheduler. For the linear probe, the SGD optimzer (Kingma & Ba, 2014) is used 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. All linear probes were trained for 100 epochs.

Hyperparameters For methods other than FroSSL, we use the CIFAR-100 hyperparameters reported in da Costa et al. (2022) on the STL-10 dataset. A batch size of 256 is used for all methods.

Hardware The backbones were trained on one NVIDIA V100 GPU.

D.3 ImageNet Experimental Details

Optimizer For the backbone, we use stochastic gradient descent (SGD) with an initial learning rate of 1e-2, weight decay of 5e-4, and a cosine annealing scheduler with warm restarts every 15 epochs. For the linear probe, the Adam optimizer is used with an initial learning rate of 5e-3, no weight decay, and a step learning rate scheduler with decreases every 10 epochs.

Epochs The backbone is trained for 100 epochs. All linear probes were trained for 100 epochs.

Hyperparameters We use λ=5e-3 for Barlow Twins as recommended in Zbontar et al. (2021). An effective batch size of 224 was used for the backbones, which equates to 56 samples per GPU. We use the same augmentation set as Chen et al. (2020).

Hardware The backbones were trained on 4 NVIDIA A100 (40GB) GPUs.

E Training Dynamics of FroSSL

E.1 Gradient with Linear Network

Given the empirical improvements to optimization speed observed in FroSSL, it is of interest to study the theoretical training dynamics of FroSSL in comparison to other SSL criterions. Inspired by the approach taken in Simon et al. (2023), we examine how FroSSL behaves in the linear network regime.

First, we use a simplified variant of FroSSL given by:

$$\mathcal{L} = ||Z_1^T Z_1 - I_d||_F^2 + ||Z_2^T Z_2 - I_d||_F^2 + ||Z_1 - Z_2||_F^2 \tag{5}$$

As compared to Equation (4), the main changes are: the mean-squared error is replaced with an equivalent formulation in terms of Frobenius norms, the logs are removed because they lead to non-linear ODEs later that are nontrivial to solve, and batchnorm is replaced with a distance to I_d. The variance terms of Equation (5) are similar to Barlow Twins, although they are defined on covariance matrices rather than cross-correlation matrices. Additionally, Equation (5) bears resemblance to a VICReg variant used to study optimal SSL representations through the lens of graph Laplacians (Balestriero & LeCun, 2022). However, the VICReg variant considers a covariance matrix of a batch containing both Z_1 and Z_2, rather than treating them individually as we do.

Second, we assume our networks are a linear mapping $W_1, W_2 \in \mathbb{R}^{d \text{x} m}$. In particular, we present a general analysis where each branch is not restricted to share weights. Because implicitly $Z_1 = f(X_1)$, we can simplify Equation (5) as

$$\mathcal{L} = ||W_1 \Gamma_1 W_1^T - I_d||_F^2 + ||W_2 \Gamma_2 W_2^T - I_d||_F^2 + ||X_1 W_1^T - X_2 W_2^T||_F^2 \tag{6}$$

where we have defined the data covariance $\Gamma_1 = X_1^T X_1 \in \mathbb{R}^{m \text{x} m}$. Because we wish to understand the training dynamics of W_1 and W_2, we define their gradients as:

$$\frac{dW_1}{dt} = \nabla_{W_1} \mathcal{L} = -2(W_1 \Gamma_1 + 2W_2 X_2^T X_1 + 2W_1 \Gamma_1 W_1^T W_1 \Gamma_1) \tag{7}$$

$$\frac{dW_2}{dt} = \nabla_{W_2} \mathcal{L} = -2(W_2 \Gamma_2 + 2W_1 X_1^T X_2 + 2W_2 \Gamma_2 W_2^T W_2 \Gamma_2) \tag{8}$$

We show how these gradients were derived in Appendix F.2. Because of the difficulty of solving Equations (7) and (8) for arbitrary choices of W, in the subsequent section we choose particular W_1 and W_2 that aids analysis.

E.2 Aligning W with the data covariance

For brevity, we describe the initialization for W_1, though W_2 follows identically. Because we train with gradient descent, we parameterize W_1 in terms of time t as $W_1(t)$. As described in Simon et al. (2023), one powerful choice of $W_1(0)$ when initializing $W_1 = W_1(0)$ is setting the right singular vectors of W_1 to be the top eigenvectors of the data covariance Γ_1. This type of initialization is called "aligned initialization". One critical assumption henceforth is that for finite batch sizes the data covariances per view, Γ_1 and Γ_2, share eigenvectors but perhaps have differing eigenvalues. This is reasonable because X_1 and X_2 are drawn from the same distribution and are augmented with the same random transforms. Next, we define the eigendecompositions of Γ_1, Γ_2, and $W_1(0)$ as:

$$\Gamma_1 = V D_1 V^T \tag{9}$$

$$\Gamma_2 = V D_2 V^T \tag{10}$$

$$W_1(0) = U_1 S_1(0) \hat{V}^{\leq d} \tag{11}$$

where $U_1 \in \mathbb{R}^{d \text{x} d}$ are arbitrary orthonormal matrices, $\hat{V}^{\leq d} \in \mathbb{R}^{d \text{x} m}$ are the top d eigenvectors of Γ_1 and Γ_2, and $S_1(0) \in \mathbb{R}^{d \text{x} m}$ are diagonal matrices of singular values such that $S_1(0) = \text{diag}(s_{1,1}(0), \cdots, s_{1,d}(0))$ with $s_{1,j}(0) > 0$. The matrices U_1 and $S_1(0)$ may be thought of as our initial random parameters, with $\hat{V}^{\leq d}$ placing constraints on the final orientation of $W(0)$. The matrices $U_2(0)$ and $S_2(0)$ are defined similarly for $W_2(0)$. The training dynamics of $W_1(0)$ from Equation (7) are given by the following proposition:

Proposition E.1 (Aligned Initialization)**.** If W_1 is initialized by (11), with W_2 being initialized similarly, then the singular vectors of W_1 do not change over time. The W_1 parameterized by t given by

$$W_1(t) = U_1 S(t) \hat{V}^{\leq d} \tag{12}$$

and the singular values evolve according to

$$s_{1,j}(t) = \sqrt{\frac{2\sqrt{\lambda_j \gamma_j} - \lambda_j}{(2\lambda_j^2 - \frac{\lambda_j - 2\sqrt{\lambda_j \gamma_j}}{s_{1,j}^2(0)}) \exp\left[8t\sqrt{\lambda_j \gamma_j} - 4\lambda_j t\right] - 2\lambda_j^2}} \tag{13}$$

Proof. A full proof is given in Appendix F.3. A quick outline is that plugging (12) into (7) gives an ordinary differential equation (ODE). Solving this ODE as an initial value problem gives (13). □

Proposition E.2 (Small Initialization)**.** If W_1, W_2 have initial weights drawn from $\mathcal{N}(0, \sigma^2)$, for sufficiently small σ, then they have the same training dynamics of aligned initialization as described in Proposition E.1. This is proven in Simon et al. (2023).

F Proofs

F.1 Sample- and Dimension-Contrastive

Here we introduce necessary definitions and propositions to show that FroSSL is both sample- and dimension-contrastive. The following two definitions are from Garrido et al. (2023b).

Definition F.1 (Dimension-Contrastive Method)**.** An SSL method is said to be dimension-contrastive if it minimizes the non-contrastive criterion $\mathcal{L}_{nc}(Z) = ||Z^T Z - \text{diag}(Z^T Z)||_F^2$, 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 F.2 (Sample-Contrastive Method)**.** An SSL method is said to be sample-contrastive if it minimizes the contrastive criterion $\mathcal{L}_c(Z) = ||ZZ^T - \text{diag}(ZZ^T)||_F^2$. This may be interpreted as penalizing the similarity between pairs of different images.

Proposition F.1. If every embedding dimension is normalized to have equal variance, then FroSSL is a dimension-contrastive method. The proof is shown in Appendix F.1.

Proof. We start with rewriting the $\arg\min$ of Equation (4) as such:

$$\begin{aligned}
\arg\min_{Z_1, Z_2} \mathcal{L}_{\text{FroSSL}} &= \arg\min_{Z_1, Z_2} \left[\log(||Z_1^T Z_1||_F^2) + \log(||Z_2^T Z_2||_F^2) + \mathcal{L}_{\text{MSE}}(Z_1, Z_2)\right] \\
&= \arg\min_{Z_1, Z_2} \left[||Z_1^T Z_1||_F^2 + ||Z_2^T Z_2||_F^2 + \mathcal{L}_{\text{MSE}}(Z_1, Z_2)\right]
\end{aligned}$$

Without loss of generality, assume that each dimension has unit variance. Then both covariance matrices have 1 in each diagonal element.

$$\begin{aligned}
&= \arg\min_{Z_1, Z_2} \left[||Z_1^T Z_1 - \text{diag}(Z_1^T Z_1)||_F^2 + ||Z_2^T Z_2 - \text{diag}(Z_2^T Z_2)||_F^2 + 2D + \mathcal{L}_{\text{MSE}}(Z_1, Z_2)\right] \\
&= \arg\min_{Z_1, Z_2} \left[\mathcal{L}_{nc}(Z_1) + \mathcal{L}_{nc}(Z_2) + 2D + \mathcal{L}_{\text{MSE}}(Z_1, Z_2)\right]
\end{aligned}$$

Thus we have that the embeddings that minimize FroSSL also minimize the non-contrastive losses $\mathcal{L}_{nc}$ for both views. □

Proposition F.2. If every embedding is normalized to have equal norm, then FroSSL is a sample-contrastive method. The proof is shown in Appendix F.1.

Proof. Using the duality of the Frobenius norm ($||A^T A||_F^2 = ||AA^T||_F^2$), we rewrite Equation (4) to use Gram matrices rather than covariance matrices:

$$\begin{aligned}
\mathcal{L}_{\text{FroSSL}} &= \log(||Z_1^T Z_1||_F^2) + \log(||Z_2^T Z_2||_F^2) + \mathcal{L}_{\text{MSE}}(Z_1, Z_2) \\
&= \log(||Z_1 Z_1^T||_F^2) + \log(||Z_2 Z_2^T||_F^2) + \mathcal{L}_{\text{MSE}}(Z_1, Z_2)
\end{aligned}$$

Assuming that each embedding is normalized to have unit norm, then both Gram matrices have 1 in each diagonal element. Then the rest of the proof then follows similarly to Proposition F.1. □

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

F.2 Derivation of FroSSL Variant Gradient

We start with Equation (6) and derive each term individually.

Term 3: The third term of (6), which corresponds to the MSE invariance term of (4), can be rewritten as:

$$||X_1W_1^T - X_2W_2^T||_F^2 = \text{trace}((X_1W_1^T - X_2W_2^T)^T(X_1W_1^T - X_2W_2^T)) \tag{14}$$
$$= \text{trace}(W_1X_1^TX_1W_1^T + W_2X_2^TX_2W_2^T - 2W_2X_2^TX_1W_1^T) \tag{15}$$
$$= \text{trace}(W_1X_1^TX_1W_1^T) + \text{trace}(W_2X_2^TX_2W_2^T) - 2\text{trace}(W_2X_2^TX_1W_1^T) \tag{16}$$

Using Equations 102 and 111 in the Matrix Cookbook (Petersen et al., 2008), we get the gradient as

$$\begin{aligned}\nabla_{W_1}||X_1W_1^T - X_2W_2^T||_F^2 &= 2W_1X_1^TX_1 - 2W_2X_2^TX_1 \\ &= 2W_1\Gamma_1 - 2W_2X_2^TX_1\end{aligned} \tag{17}$$

$$\begin{aligned}\nabla_{W_2}||X_1W_1^T - X_2W_2^T||_F^2 &= 2W_2X_2^TX_2 - 2W_1X_1^TX_2 \\ &= 2W_2\Gamma_2 - 2W_1X_1^TX_2\end{aligned} \tag{18}$$

Term 1: The first term of (6), which corresponds to the argument of the View 1 logarithm of (4), is derived using Equation 6 of Simon et al. (2023). In particular, we get

$$\nabla_{W_1}||W_1\Gamma_1W_1^T - I_d||_F^2 = -4(W_1\Gamma_1W_1^T - I_d)W_1\Gamma_1 \tag{19}$$
$$\nabla_{W_2}||W_1\Gamma_1W_1^T - I_d||_F^2 = 0 \tag{20}$$

Term 2: The second term of (6) follows similarly to the first term.

$$\nabla_{W_1}||W_2\Gamma_2W_2^T - I_d||_F^2 = 0 \tag{21}$$
$$\nabla_{W_2}||W_2\Gamma_2W_2^T - I_d||_F^2 = -4(W_2\Gamma_1W_2^T - I_d)W_2\Gamma_2 \tag{22}$$

Combining Everything: We can now combine all of our gradients to find (7) and (8).

$$\begin{aligned}\nabla_{W_1}\mathcal{L} &= 2W_1\Gamma_1 - 2W_2X_2^TX_1 - 4(W_1\Gamma_1W_1^T - I_d)W_1\Gamma_1 \\ &= -2(W_1\Gamma_1 + 2W_2X_2^TX_1 + 2W_1\Gamma_1W_1^TW_1\Gamma_1)\end{aligned} \tag{23}$$

$$\begin{aligned}\nabla_{W_2}\mathcal{L} &= 2W_2\Gamma_2 - 2W_1X_1^TX_2 - 4(W_2\Gamma_2W_2^T - I_d)W_2\Gamma_2 \\ &= -2(W_2\Gamma_2 + 2W_1X_1^TX_2 + 2W_2\Gamma_2W_2^TW_2\Gamma_2)\end{aligned} \tag{24}$$

F.3 Derivation of Linear layer Gradient w.r.t Time

Plugging (12) into (7), we immediately get

$$\begin{aligned}\frac{dW_1}{dt} = 2US_1(t)D_1\hat{V}^{\odot\leq d} - 4US_1(t)(D_1^{\odot\frac{1}{2}} \odot D_2^{\odot\frac{1}{2}})\hat{V}^{\leq d} \\ - 4US_1^3(t)D_1^2\hat{V}^{\leq d}\end{aligned} \tag{25}$$

Next, one should recognize that all terms are left-multiplied by U and right-multiplied by $\hat{V}^{\leq d}$. This lets us combine everything into a more compact form.

$$\frac{dW}{dt} = 2U\left[S_1(t)\left(D_1 - 2(D_1^{\odot\frac{1}{2}} \odot D_2^{\odot\frac{1}{2}}) - 2S_1^2(t)D_1^2\right)\right]\hat{V}^{\leq d} \tag{26}$$

It can be seen in (26) that the singular vectors of W remain unchanged over time. The only changes are the singular values which are given in the brackets. The dynamics of the singular values over time constitute an ODE given by

$$s'_{1,j}(t) = 2s_{1,j}(t)\left(\lambda_j - 2\sqrt{\lambda_j\gamma_j} - 2s^2_{1,j}(t)\lambda_j^2\right) \tag{27}$$

where λ_j, γ_j are the j-th largest eigenvalues of D_1, D_2, respectively. One can find the general solution to this ODE using their favorite ODE solver. It gives a solution in the form:

$$s_{1,j}(t) = \sqrt{\frac{2\sqrt{\lambda_j\gamma_j} - \lambda_j}{\exp\left[2(2\sqrt{\lambda_j\gamma_j} - \lambda_j)(c_1 + 2t)\right] - 2\lambda_j^2}} \tag{28}$$

where c_1 is some constant. We can find c_1 by solving the initial value problem given by our initial S_1 matrix. Thus (28) can be rewritten as:

$$s_{1,j}(t) = \sqrt{\frac{2\sqrt{\lambda_j\gamma_j} - \lambda_j}{(2\lambda_j^2 - \frac{\lambda_j - 2\sqrt{\lambda_j\gamma_j}}{s^2_{1,j}(0)})\exp\left[8t\sqrt{\lambda_j\gamma_j} - 4\lambda_j t\right] - 2\lambda_j^2}} \tag{29}$$