On the duality between contrastive and non-contrastive self-supervised learning

Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, Yann Lecun

On the duality between contrastive and non-contrastive self-supervised learning

Quentin Garrido[1,2*] **Yubei Chen**[1] **Adrien Bardes**[1,3] **Laurent Najman**[2]

Yann LeCun[1,4,5]

[1]Meta AI
[2]Université Gustave Eiffel, CNRS, LIGM, F-77454 Marne-la-Vallée, France
[3]Inria, École normale supérieure, CNRS, PSL Research University
[4]Courant Institute, New York University
[5]Center for Data Science, New York University

Abstract

Recent approaches in self-supervised learning of image representations can be categorized into different families of methods and, in particular, can be divided into contrastive and non-contrastive approaches. While differences between the two families have been thoroughly discussed to motivate new approaches, we focus more on the theoretical similarities between them. By designing contrastive and non-contrastive criteria that can be related algebraically and shown to be equivalent under limited assumptions, we show how close those families can be. We further study popular methods and introduce variations of them, allowing us to relate this theoretical result to current practices and show how design choices in the criterion can influence the optimization process and downstream performance. We also challenge the popular assumptions that contrastive and non-contrastive methods, respectively, need large batch sizes and output dimensions. Our theoretical and quantitative results suggest that the numerical gaps between contrastive and non-contrastive methods in certain regimes can be significantly reduced given better network design choice and hyperparameter tuning.

1 Introduction

Self-supervised learning (SSL) of image representations has shown significant progress in the last few years [7, 17, 8, 14, 21, 5, 34, 2, 30, 6, 10, 22, 36, 37, 15], approaching, and sometime even surpassing, the performance of supervised baselines on many downstream tasks. Most recent approaches are based on the joint-embedding framework with a siamese network architecture [3] which are divided into two main categories, contrastive and non-contrastive methods. Contrastive methods bring together embeddings of different views of the same image while pushing away the embeddings from different images. Non-contrastive methods also attract embeddings of views from the same image but remove the need for explicit negative pairs, either by architectural design [14, 9] or by regularization of the variance and covariance of the embeddings [34, 2, 24].

While contrastive and non-contrastive approaches seem very different and have been described as such [34, 2, 12, 14], we propose to take a closer look at the similarities between the two, both from a theoretical and empirical point of view and show that there exists a close relationship

*Correspondence to garridoq@fb.com

Preprint. Under review.

between them. We focus on covariance regularization-based non-contrastive methods [34, 12, 2] and demonstrate that these methods can be seen as contrastive between the dimensions of the embeddings instead of contrastive between the samples. We, therefore, introduce the term *dimension-contrastive* methods which we believe is better suited for them and refer to the original contrastive methods as *sample-contrastive* methods. To show the similarities between the two, we define contrastive and non-contrastive criteria based on the Frobenius norm of the Gram and covariance matrices of the embeddings, respectively, and show the equivalence between the two under assumptions related to the type of normalization performed on the embeddings. We then relate popular methods to these criteria, highlighting the links between them and further motivating the use of the *sample-contrastive* and *dimension-contrastive* nomenclature.

Finally, we introduce variations of an existing non-contrastive method VICReg, and a contrastive one, SimCLR, allowing us to experimentally study the links between sample-contrastive and dimension-contrastive methods, providing experimental insights on how to design and train these criteria.

Our contributions can be summarized as follows:

- We define two non-contrastive and contrastive criteria and show that they are equal up to a constant for certain normalizations of the embeddings. By relating popular methods to them, we show how close sample-contrastive and dimension-contrastive methods are.
- We introduce methods that interpolate between VICReg and SimCLR to study the impact of precise components of their loss functions.
- We study how well the criteria are optimized in popular methods and show that some design choices such as using an InfoNCE-based criterion impact negatively this process.
- We show that all else being equal, a dimension-contrastive criterion can lead to better downstream performance than its sample-contrastive counterpart. We demonstrate the importance of the projector's architecture on performance, and how a better design improves robustness to the embedding dimension, significantly improving known performance.

2 Related work

Contrastive methods. In self-supervised learning of image representations, contrastive methods pull together embeddings of distorted views of a single image while pushing away embeddings coming from different images. Many works in this direction have recently flourished [7, 17, 8, 10, 32], most of them using the InfoNCE criterion [25], except [15], that uses squared similarities between the samples. Clustering-based methods [4, 5, 6] can be seen as contrastive between prototypes, or clusters, instead of samples.

Non-contrastive methods. Recently, methods that deviate from contrastive learning have emerged and eliminate negative samples in different ways. Distillation based methods such as BYOL [14], SimSiam [9] or DINO [6] use architectural tricks inspired by distillation to avoid the collapse problem. Information maximization methods [2, 34, 12, 24] maximize the informational content of the representations and have also had significant success. They rely on regularizing the empirical covariance matrix of the embeddings such that their informational content is maximized. Our study of non-contrastive learning will focus on these covariance-based methods.

Understanding contrastive and non-contrastive learning. Recent works tackle the task of understanding and characterizing methods. The fact that a method like SimSiam does not collapse is studied in [29]. The loss landscape of SimSiam is also compared to SimCLR's in [26], which shows that it learns bad minima. In [31], the optimal solutions of the InfoNCE criterion are characterized, giving a better understanding of the embedding distributions. A spectral graph point of view is taken in [16, 15, 27] to analyze self-supervised learning methods. In [1], popular self-supervised methods are linked to spectral methods, providing a unifying framework that highlights differences between them. The gradient of various methods is also studied in [28], where they show links and differences between them.

3 Equivalence of the contrastive and non-contrastive criterion

While our results only depend on the embeddings and not the architecture used to obtain then, nor do they depend on the data modality, all the studied methods are placed in a joint embedding framework and applied on images. Given a dataset $\mathcal{D}$ with individual datum $d_i \in \mathbb{R}^{c \times h \times w}$, this datum is augmented to obtain two views x_i and x'_i. These two views are then each fed through a pair of neural networks f_θ and $f'_{\theta'}$. We obtain the *representations* $f_\theta(x_i)$ and $f'_{\theta'}(x'_i)$, which are fed through a pair of projectors p_θ and $p'_{\theta'}$ such that *embeddings* are defined as $p_\theta(f_\theta(x_i))$ and $p'_{\theta'}(f'_{\theta'}(x'_i))$. We denote the matrices of embeddings $\mathcal{K}$ and $\mathcal{K}'$ such that $\mathcal{K}_{.,i} = p_\theta(f_\theta(x_i))$, and similarly for $\mathcal{K}'$, we have $\mathcal{K} \in \mathbb{R}^{M \times N}$, with M the embedding size and N the batch size, and similarly for $\mathcal{K}'$. These embedding matrices are the primary object of our study. In practice, we use $f_\theta = f'_{\theta'}$ and $p_\theta = p'_{\theta'}$. While most self-supervised learning approaches use positive pairs (x_i, x'_i) and negative pairs $\{\forall j, j \neq i, (x_i, x_j)\} \bigcup \{\forall j, j \neq i, (x_i, x'_j)\}$ for a given view x_i, we focus on the simpler scenario where negative samples are just $\{\forall j, j \neq i, (x_i, x_j)\}$. There is no fundamental difference when $\theta = \theta'$ and when the same distribution of augmentations is used for both branches, and we therefore make these simplifications to make the analysis less convoluted.

We start by defining precisely which contrastive and non-contrastive criteria we will be studying throughout this work. These criteria will be used to classify methods in two classes, *sample-contrastive*, which corresponds to what is traditionally thought of as contrastive, and *dimension-contrastive*, which will encompass non-contrastive methods relying on regularizing the covariance matrix of embeddings. While we focus on the repulsive force, it is worth noting that these criteria are not optimized alone. They are usually combined with an invariance criterion that aims at producing the same representation for two views of the same image. This invariance criterion is generally a similarity measure such as the cosine similarity or the mean squared error of the difference between a positive pair of samples. Both are equivalent from an optimization point of view if using normalized embeddings.

Definition 3.1. Given a matrix $A \in \mathbb{R}^{n \times n}$. We define its *extracted diagonal* $\text{diag}\,(A) \in \mathbb{R}^{n \times n}$ as:

$$\text{diag}\,(A)_{i,j} = \begin{cases} A_{i,i}, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Definition 3.2. A method is said to be *sample-contrastive* if it minimizes the contrastive criterion $L_c = \|\mathcal{K}^T\mathcal{K} - \text{diag}(\mathcal{K}^T\mathcal{K})\|_F^2$. Similarly, a method is said to be *dimension-contrastive* if it minimizes the non-contrastive criterion $L_{nc} = \|\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\|_F^2$.

The *sample-contrastive* criterion can be seen as penalizing the similarity between different pairs of images, whereas the *dimension-contrastive* criterion can be seen as penalizing the off-diagonal terms of the covariance matrix of the embeddings. These criteria respectively try to make pairs of samples or dimensions orthogonal. We also define a generalization of the *sample-contrastive* criterion that can be applied to certain methods that do not explicitly make the embeddings orthogonal.

Definition 3.3. Considering access to an infinite amount of augmented views, and thus negative pairs (x, x^-), a method is said to be *weakly-sample-contrastive* if it minimizes the following generalization of the contrastive criterion, as the dimension of embeddings M goes to infinity:

$$L_{wc} = \mathbb{E}_{(x,x^-)}\left[\left(x^T x^-\right)^2\right]. \tag{2}$$

Now that we have properly defined the criteria that we are considering, we will classify various popular contrastive and non-contrastive methods.

Proposition 3.1. DCL [32] and SimCLR [7] are weakly-sample-contrastive.

The main reason why SimCLR and DCL cannot be easily linked to L_c but only to $\mathcal{L}_{wc}$ comes from the use of their cosine similarities instead of their square or absolute value. Indeed, while our criteria aim at making pairs of embeddings or dimensions orthogonal, SimCLR and DCL's criteria go a step further and aim at making them opposite. Both cannot be satisfied perfectly in practice, as we would need as many dimensions as samples for our criterion to be perfectly satisfied, and more than two vectors cannot be pairwise opposite for SimCLR and DCL's criterion. Since we are interested in how the links between methods manifest themselves in practice, we introduce SimCLR-sq and SimCLR-abs as variations of SimCLR, which respectively use square or absolute values of cosine

similarities. We define DCL-sq and DCL-abs similarly. We provide a study of SimCLR-sq and SimCLR-abs in supplementary section D, where we compare them to SimCLR. The main conclusion is that the distribution of off-diagonal terms of the Gram matrix is similar between all studied methods, with a high concentration of values around zero, and that changing SimCLR into these variations does not impact performance. We even see an increase in top-1 accuracy on ImageNet [11] with linear evaluation when using SimCLR-abs, where we reach 66.81% accuracy, compared to 66.33% with our reproduction of SimCLR.

Proposition 3.2. SimCLR-abs/sq, DCL-sq/abs, and Spectral Contrastive Loss [15] are sample-contrastive methods. Barlow Twins [34], VICReg [2] and TCR [24] are dimension-contrastive methods.

From propositions 3.1 and 3.2 we can see that sample-contrastive and dimension-contrastive methods can respectively be linked together by L_c and L_{nc}. This alone is not enough to show the link between those two families of methods and we will now discuss the link between L_c and L_{nc} to show how close those families are.

Theorem 3.3. The sample-contrastive and dimension-contrastive criteria L_c and L_{nc} are equivalent up to row and column normalization of the embedding matrix $\mathcal{K}$. Consider a batch size of N and an embedding dimension of M. We have:

$$L_{nc} + \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 = L_c + \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4. \tag{3}$$

Theorem 3.3 is similar to lemma 3.2 from [20], where we consider matrices that are not doubly stochastic. It is worth noting that our result does not rely on any assumption about the embeddings themselves. A similar result was also used recently in [16], where they relate the spectral contrastive loss to L_{nc}.

The proof of theorem 3.3 hinges on the fact that the squared Frobenius norm of the Gram and Covariance matrix of the embeddings are equal, i.e., $\|\mathcal{K}^T\mathcal{K}\|_F^2 = \|\mathcal{K}\mathcal{K}^T\|_F^2$. This means that penalizing all the terms of the Gram matrix (i.e., pairwise similarities) is the same as penalizing all of the terms of the Covariance matrix. While this gives an intuition for the similarity between the contrastive and non-contrastive criteria, it is not as representative of the criteria used in practice as L_c and L_{nc} are.

While theorem 3.3 shows that sample-contrastive and dimension-contrastive approaches minimize similar criteria, for none of these methods can we conclude that both criteria can be used interchangeably. However, if both rows and columns of $\mathcal{K}$ were L2 normalized, we would have $L_{nc} = L_c + N - M$. In this case, both criteria would be equivalent from an optimization point of view, and we could conclude that sample-contrastive and dimension-contrastive methods are all minimizing the same criterion. Doubly normalizing the embedding matrix has been explored for VICReg, where the dimensions were normalized via the variance criterion and the embeddings are l2 normalized, confer table 8 in [2]. While this leads to a drop in top-1 accuracy on ImageNet of 3.5 points, this still leads to performances similar to SimCLR in a scenario where we know that the sample-contrastive and dimensions-contrastive criteria are equal up to a constant.

Influence of normalization. The difference between the two criteria then lies in the embedding matrix row and column norms, and most approaches do normalize it in one direction. Since SimCLR relies on the cosine distance as a similarity measure between embeddings, we can effectively say that it uses normalized embeddings. Similarly, Spectral Contrastive Loss projects the embeddings on a ball of radius $\sqrt{\mu}$, with μ a tuned parameter, meaning that the embeddings are normalized before the computation of the loss function.
Barlow Twins normalizes dimensions such that they have a null mean and unit variance, so all dimensions will have a norm of $\sqrt{N}$. VICReg takes a similar approach where dimensions are centered, but their variance is regularized by the variance criterion. This is very similar to what is done for Barlow Twins and thus leads to dimensions with constant norm. However, for TCR, the embeddings are normalized and not the dimensions, contrasting with other covariance based methods. One of the main differences between normalizing embeddings or dimensions is that in the former case, embeddings are projected on a $M-1$ dimensional hypersphere, and in the latter, they are not constrained on a particular manifold; instead, their spread in the ambient space is limited.

Nonetheless, a constraint on the norm of the embeddings also constrains the norm of the dimensions indirectly, and vice versa, as illustrated in lemma 3.4.

Lemma 3.4. If embeddings are normalized such that $\forall i,\ \|\mathcal{K}_{\cdot,i}\|_2 = a$ we have

$$\frac{N^2}{M}a^4 \leq \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 \leq N^2 a^4. \tag{4}$$

Conversely, if dimensions are normalized such that $\forall j,\ \|\mathcal{K}_{j,\cdot}\|_2 = a$ we have

$$\frac{M^2}{N}a^4 \leq \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4 \leq M^2 a^4. \tag{5}$$

Following the proof of lemma 3.4, the lower bounds can be constructed with a constant embedding matrix and the upper bounds with an embedding matrix where either the rows or columns contain only one non-zero element. Both correspond to collapsed representations and will thus not be attained in practice. While it is impossible to characterize non-collapsed embedding matrices and, as such, derive better practical bounds, these bounds can still be useful to derive the following corollary. We study how close methods are to these bounds in practice in section E of the supplementary material. The main conclusion is that in all practical scenarios, the sum of norms will be very close to the lower bounds, deviating by a single-digit factor.

Corollary 3.4.1. If embeddings are L2-normalized we have

$$L_{nc} - N + \frac{N^2}{M} \leq L_c \leq L_{nc} - N + N^2. \tag{6}$$

Similarly, if dimensions are L2-normalized we have

$$L_c - M + \frac{M^2}{N} \leq L_{nc} \leq L_c - M + M^2. \tag{7}$$

Lemma 3.4 applied to Theorem 3.3 directly gives us corollary 3.4.1, which means that in practical scenarios, even when we compare methods where the embeddings are not doubly normalized, the contrastive and non-contrastive criteria can't be arbitrarily far apart. Considering the previous discussions, we thus argue that the main differences between sample-contrastive and dimension-contrastive methods come from the optimization process as well as the implementation details.

Disguising VICReg as a contrastive method. To illustrate theorem 3.3 we can rewrite VICReg's criterion to make L_c appear. We first recall the different components of VICReg's criterion. The variance criterion v is a hinge loss that aims at making the variance along every dimension greater than 1, and the covariance criterion c is exactly defined as L_{nc} applied to centered embeddings. For more details, confer [2]. To make L_c appear, we will still apply the invariance and variance criterion on the embeddings, but the covariance criterion will be applied to the transposed embeddings, effectively making it contrastive since we have:

$$\mathrm{c}(\mathcal{K}^T) = \|\mathcal{K}^T \left(\mathcal{K}^T\right)^T - \mathrm{diag}\left(\mathcal{K}^T \left(\mathcal{K}^T\right)^T\right)\|_F^2 = \|\mathcal{K}^T\mathcal{K} - \mathrm{diag}(\mathcal{K}^T\mathcal{K})\|_F^2 = L_c(\mathcal{K}). \tag{8}$$

We then just need to add a regularization term on the norms of embeddings and dimensions as follows:

$$L_{reg}(\mathcal{K}) = \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4 - \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4,$$

and VICReg's loss function can then be written as

$$\mathcal{L}_{VICReg} = \lambda \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i} - \mathcal{K}'_{\cdot,i}\|_2^2 + \mu\left(v(\mathcal{K}) + v(\mathcal{K}')\right) + \nu\left(L_c(\mathcal{K}) + L_{reg}(\mathcal{K}) + L_c(\mathcal{K}') + L_{reg}(\mathcal{K}')\right). \tag{9}$$

Being able to make VICReg's criterion sample-contrastive highlights the close relationship between sample-contrastive and dimension-contrastive methods and further shows that the difference in the behavior of different methods is not mainly due to whether they are contrastive or not.

Implications for further analysis. While we have shown that the contrastive and non-contrastive criteria are closely related and even equivalent when doubly normalizing the embedding matrix, all formulations are not as easy to work with for theoretical analysis. In [16], the criterion $\mathcal{L}_\sigma$ is very close to VICReg's criterion, with the variance criterion implicitly defined through the use of the identity matrix in their regularizer. The use of this criterion and its link to Spectral Contrastive Loss' allowed them to analyze more easily such methods. We hope that theoretical analyses of self-supervised learning focusing on the optimization of the criterion will be able to apply to a larger category of methods through theorem 3.3, which can be used both to link methods together as well as derive formulations that are easier to work with.

4 Quality of the optimization and its transferability from the embeddings to the representations.

While we have discussed the link between the contrastive and non-contrastive criteria, they are optimized on the embeddings and not on the representations, which are used in practice for evaluation. One can also wonder how the design differences in popular criteria affect the quality of this optimization process. We will thus focus on these two points, and we start by introducing variations on VICReg that will allow us to interpolate between VICReg and SimCLR while isolating precise components of the loss function.

VICReg variations. We introduce two variants of VICReg, one that is non-contrastive but inspired by the InfoNCE criterion and one that is contrastive and also inspired by the InfoNCE criterion. The former is motivated by one of the main differences between methods, which is the use of the LogSumExp (LSE) for the repulsive force (e.g., SimCLR) or the use of the sum of squares (e.g., SCL, VICreg, BT). The latter is motivated by the wish to design contrastive methods, where implementation details such as the negative pair sampling are as close as possible to another method. This way, comparing VICReg to either of those methods will yield a comparison that truly isolates specific components of the loss function. These two methods can also be seen as a transformation from VICReg to SimCLR, which allows us to see when the behavior of VICReg becomes akin to SimCLR's, as illustrated in the following diagram:

$$\text{VICReg} \xrightarrow{\text{LogSumExp}} \text{VICReg-exp} \xrightarrow{\text{Contrastive}} \text{VICReg-ctr} \xrightarrow{\text{Neg. pair sampling}} \text{SimCLR}$$

The first variant that we will introduce is VICReg-exp, which uses a repulsive force inspired by the InfoNCE criterion. We first define the exponential covariance regularization as:

$$c_{exp}(\mathcal{K}) = \frac{1}{d} \sum_i \log \left(\sum_{j \neq i} e^{C(\mathcal{K})_{i,j}/\tau} \right), \tag{10}$$

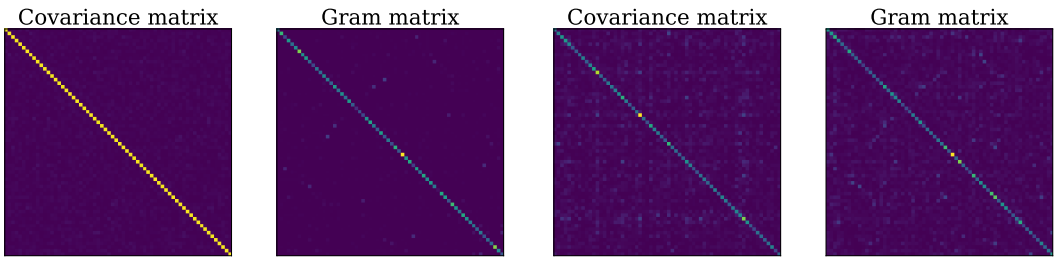
VICReg-exp is then VICReg where we replace the covariance criterion by this exponential covariance criterion, giving an overall criterion of

$$\mathcal{L}_{VICReg-exp} = \lambda \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i} - \mathcal{K}'_{\cdot,i}\|_2^2 + \mu \left(v(\mathcal{K}) + v(\mathcal{K}') \right) + \nu \left(c_{exp}(\mathcal{K}) + c_{exp}(\mathcal{K}') \right). \tag{11}$$

We then define VICReg-ctr, which is VICReg-exp where we transpose the embedding matrix before applying the variance and covariance regularization. This means that the variance regularization will regularize the norm of the embeddings, and the covariance criterion now penalizes the Gram matrix, with the same repulsive force as in DCL. Transposing the embedding matrix for the variance criterion leads to more stable training, enables the use of mixed precision, and has little to no influence on performance compared to keeping the original variance criterion. We study the difference between the two strategies in the supplementary section F. We thus have the following criterion:

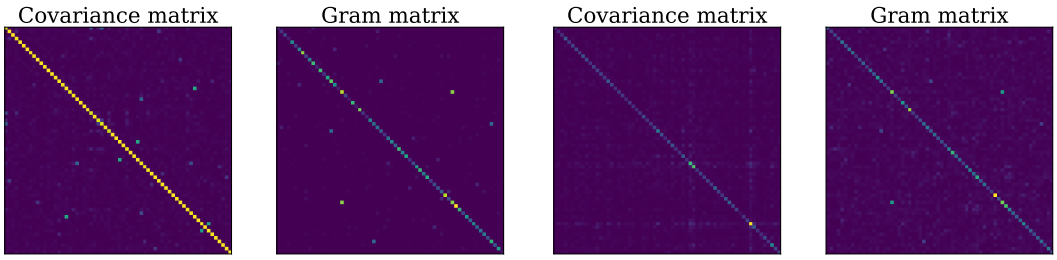
$$\mathcal{L}_{VICReg-ctr} = \lambda \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i} - \mathcal{K}'_{\cdot,i}\|_2^2 + \mu \left(v(\mathcal{K}^T) + v(\mathcal{K}'^T) \right) + \nu \left(c_{exp}(\mathcal{K}^T) + c_{exp}(\mathcal{K}'^T) \right). \tag{12}$$

This way, VICReg-exp will allow us to study the influence of the use of the LogSumExp operator in the repulsive force, and VICReg-ctr to study the difference between sample-contrastive and



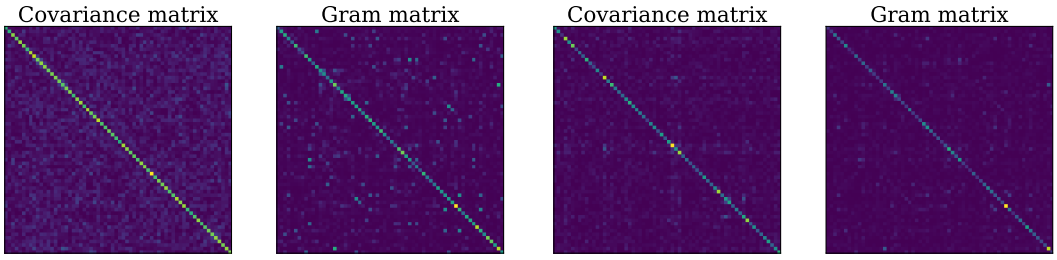
(i) On the embeddings (ii) On the representations

(a) VICReg



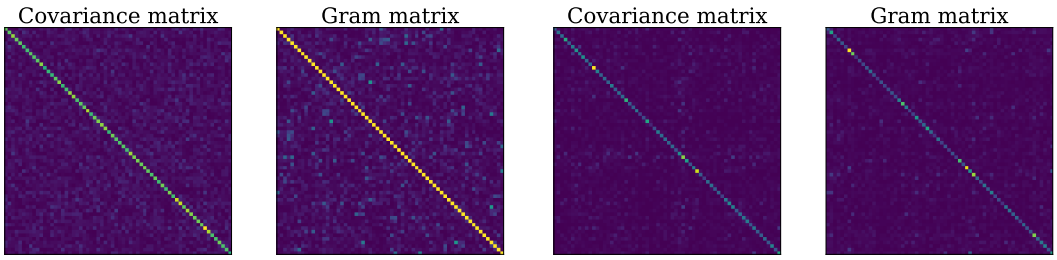
(i) On the embeddings (ii) On the representations

(b) VICReg-exp



(i) On the embeddings (ii) On the representations

(c) VICReg-ctr



(i) On the embeddings (ii) On the representations

(d) SimCLR

Figure 1: Covariance and similarity matrices on a random part of the ImageNet training set, using VICReg, VICReg-exp, VICReg-ctr, and SimCLR pretrained on ImageNet for 100 epochs. The covariance matrix is limited to the first 64 dimensions, while the Gram matrix is limited to the first 64 samples. In all cases, we used a projector with an output dimension of 2048, the same as the representation dimension.

dimension-contrastive methods when comparing it to VICReg-exp. We will now be able to study the optimization of the contrastive and non-contrastive criterion and see how different design choices affect it.

Optimization of the contrastive and non-contrastive criterion. While a perfect optimization of the aforementioned criteria would lead to embeddings with similar properties for the covariance and Gram matrix, one can wonder how well they are optimized in practice and whether design choices have a significant impact. To this effect we will look at the Gram and Covariance matrices after optimization, both on the embeddings to study the quality of the optimization process and on the representations to study the transferability of this process to the representations since they are used for downstream tasks. For the embeddings, we use the same normalization process as is used during training, and we center the representations to alleviate the fact that the last ReLU layer constrains them to the positive orthant. This centering on the representations is only done to make the visualization more interpretable.

As we can see in figure 1, while VICReg penalizes the off-diagonal terms of the covariance matrix and not the Gram matrix, both matrices have off-diagonal terms that are significantly smaller than their diagonal counterparts. Similarly for VICReg-exp, we can see that both the Gram and covariance matrices are dominated by their diagonal in the embedding space, though there is noise in the off-diagonal terms. This is due to the use of the LogSumExp, which as a smooth approximation of the max operator, will mostly penalize the largest values. On the other hand, using squared values will make them penalized by their absolute and not relative value.
We also observe the same behavior for VICReg-ctr and SimCLR which both lead to Gram and covariance matrices that are dominated by their diagonal but that are overall noisier than for VICReg and VICReg-exp. This suggests that the main culprit of this noise is indeed the LogSumExp but that the sample-contrastive nature of VICReg-ctr and SimCLR also played a role in creating it. While we only show 64 samples or dimensions here, we provide results for more in section I of the supplementary material.

Looking at the representations, the differences between the methods start to fade. They all still produce Gram and covariance matrices that are dominated by their diagonal, but with some off-diagonal noise. Even though we could see a clear difference in the quality of the optimization in the embedding space, the similarity in the representation space makes it harder to see if there will be a direct impact on downstream performance when evaluating the representations. To investigate this, we will look at how this sample-contrastive and dimension-contrastive duality manifests itself in downstream performances, with a focus on linear classification on ImageNet.

5 Practical differences between contrastive and non-contrastive methods

While we have discussed how close sample and dimension contrastive methods are in theory and that even with differences in the quality of the optimization process, the impact on representations is unclear, one of the primary considerations when choosing or designing a method is the performance on downstream tasks. Linear classification on ImageNet has been the main focus in most SSL methods, so we will focus on this task. We will consider the two following aspects, which are responsible for most of the discrepancies between methods.

Loss implementation. Thanks to VICReg-exp, we are able to study the difference between penalizing the Frobenius norm directly and using a LogSumExp to penalize it. Similarly, for VICReg-ctr we are able to study the practical differences between the contrastive and non-contrastive criteria. Finally, with SimCLR we will be able to see how the last details between VICReg-ctr and it can impact performance.

Projector architecture. One of the main differences in methods is how the projector is designed. To describe projector architectures we use the following notation: $X - Y - Z$ means that we use linear layers of dimensions X, then Y and Z. Each layer is followed by a ReLU activation and a batch normalization layer. The last layer has no activation, batch normalization, or bias.
In order to study the impact that this has on performance with respect to embedding size, we study three scenarios. First, $d - d - d$, which is the scenario used for VICReg and BT, then $2048 - d$ which was originally used for SimCLR, and finally $8192 - 8192 - d$ which was optimal for large embeddings with VICReg.

Online linear probing. Due to the extensive nature of the following experiments, we use a proxy of the classical linear evaluation on ImageNet, where the classifier is trained alongside the backbone and projector. Representations are fed to a linear classifier while keeping the gradient of this classifier's criterion from flowing back through the backbone. The addition of this linear classifier is extremely

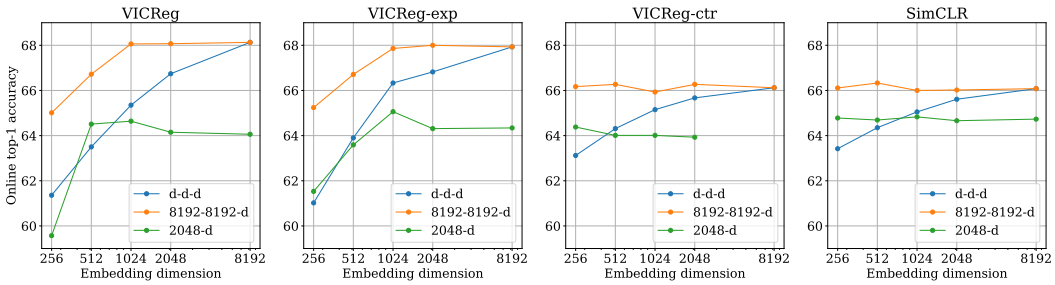


Figure 2: Online performance on ImageNet for VICReg, VICReg-exp, VICReg-ctr, and SimCLR with respect to embedding dimensions when changing the projector's architecture. Confer supplementary section H for numerical values and hyperparameters.

cheap and avoids a costly linear evaluation after training. The performance of this online classifier correlates almost perfectly with its offline counterpart, so we can rely on it to discuss the general behaviors of various methods. This evaluation was briefly mentioned in [7] but without experimental support. We discuss the correlation between the two further in the supplementary section C.

Results. The first takeaway from figure 2 is that the transition VICReg $\rightarrow$ VICReg-exp via the addition of the LogSumExp did not alter overall performance or behavior. While small performance differences are visible between the two when using light projectors, especially at low embedding dimension, as soon as we use a larger projector these differences disappear with them achieving 68.13% and 68.00% respectively. The story is similar when comparing VICReg-ctr and SimCLR. Even though VICReg-ctr uses fewer negative pairs since both branches are treated individually, the behavior and performance of both are very similar. VICReg-ctr performs around 1 point lower with a $2048-d$ projector, but with a larger one, the difference between the two disappears with them achieving 66.27% and 66.33% respectively.
Focusing on the transition VICReg-exp $\rightarrow$ VICReg-ctr, we can see a drop of around 2 points in best performance, dropping from 68.00% to 66.27%. While we have shown the similarity between the contrastive and non-contrastive criteria, they are still not perfectly equal in practical scenarios. Whether this drop is due to the difference between the two criteria, which is related to the embeddings and dimensions norms, or to issues with the optimization process is unclear. This would suggest that all else being equal, a non-contrastive criterion can achieve better downstream performance than its contrastive counterpart.

A larger projector increases performance. From figure 2 we can see that for every studied method, going from a projector with architecture $2048-d$ to $8192-8192-d$ yielded a significant boost in performance, especially for VICReg and VICReg-ctr, both gaining $3.5-4$ points. The projector $d-d-d$ is in between the two depending on the embedding dimension but also shows a similar trend, the performance increases with the number of parameters for every method. While out of the scope of this work, the study of the importance of the projector's capacity is an exciting line of work that should help gain a deeper understanding of its role in self-supervised learning. We provide a preliminary discussion in the supplementary section G.

Clearing up misconceptions. While contrastive methods are often thought of as sample inefficient, thus requiring large batch sizes, and non-contrastive methods as dimension inefficient, thus requiring projectors with large output dimensions, we argue that both of these assumptions are misleading and that all of these apparent issues can be alleviated with some care. Most notably, the need for large batch sizes of contrastive methods has been studied in [32] and [35] where the main conclusions are that with some tuning of the InfoNCE parameters the robustness of SimCLR and MoCo to small batches can be improved. Regarding the robustness of non-contrastive methods to embedding dimension, our experiments show that with a more adequate projector architecture and with careful hyperparameter tuning, the drop in performance at low embedding dimension is not as present as initially reported [34, 2]. With 256-dimensional embeddings, we were able to achieve 61.36% top-1 accuracy by tuning VICReg's hyperparameters, compared to the 55.9% that were initially reported in [2]. This can be further improved to 65.01% by using a bigger projector. While a drop is still present, we are able to reach peak performance at 1024 dimensions, which is lower than the

representation's dimension of 2048 and shows that a large embedding dimension is not a deciding factor in downstream performance.

6 Conclusion

Through an analysis of their criterion, we were able to show that sample-contrastive and dimension-contrastive methods have learning objectives that are closely related, as they are effectively minimizing criteria that are equivalent up to row and column normalization of the embedding matrix. This suggests a certain duality in the behavior of such methods, which we studied empirically. Through the lens of variations of VICReg, we were able to study popular design choices in self-supervised loss functions and show how they affect downstream performance, significantly improving the robustness to embedding dimension of VICReg along the way. All else being equal, the dimension-contrastive version of a criterion yielded better performance than its sample-contrastive counterpart, raising questions as to how the minor differences between the two play a role in the representation quality. We expect that our results will help extend theoretical works in self-supervised learning to a wider family of methods and help alleviate preconceived ideas on contrastive and non-contrastive learning.

References

[1] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022. 2

[2] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1, 2, 4, 5, 9

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sackinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *NeurIPS*, 1994. 1

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning. In *ECCV*, 2018. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1, 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, and Julien Mairal Piotr Bojanowski Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 1, 2, 3, 9

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

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2020. 1, 2

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 1, 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[12] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning, 2021. 1, 2

[13] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022. 12

[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. In *NeurIPS*, 2020. 1, 2, 15

[15] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *NeurIPS*, 34, 2021. 1, 2, 4

[16] Jeff Z HaoChen, Colin Wei, Ananya Kumar, and Tengyu Ma. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. *arXiv preprint arXiv:2204.02683*, 2022. 2, 4, 6

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 15

[19] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. 16

[20] Quoc Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Ng. Ica with reconstruction cost for efficient overcomplete feature learning. *NeurIPS*, 24, 2011. 4, 14

[21] Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive visual representations. In *NeurIPS*, 2021. 1

[22] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. In *ICLR*, 2022. 1

[23] Shengqiao Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011. 12

[24] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022. 1, 2, 4, 14

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[26] Ashwini Pokle, Jinjin Tian, Yuchen Li, and Andrej Risteski. Contrasting the landscape of contrastive and non-contrastive learning. *arXiv preprint arXiv:2203.15702*, 2022. 2

[27] Kendrick Shen, Robbie Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. *arXiv preprint arXiv:2204.00570*, 2022. 2

[28] 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. *arXiv preprint arXiv:2112.05141*, 2021. 2

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

[30] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022. 1

[31] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939. PMLR, 2020. 2, 12

[32] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021. 2, 3, 9

[33] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 15

[34] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pages 12310–12320. PMLR, 2021. 1, 2, 4, 9

[35] Chaoning Zhang, Kang Zhang, Trung X Pham, Axi Niu, Zhinan Qiao, Chang D Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. *arXiv preprint arXiv:2203.17248*, 2022. 9

[36] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 1

[37] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. 2022. 1

A Proofs

We start by discussing two useful Lemmas related to weakly-sample-contrastive methods.

Lemma A.1. Let $X, Y \sim \sigma^{D-1}$ two i.i.d. random variables corresponding to vectors uniformly distributed on S^{D-1}. Their dot product follows the following distribution

$$\frac{X^T Y + 1}{2} \sim \text{Beta}\left(\frac{D-1}{2}, \frac{D-1}{2}\right).$$

Proof. A similar result was proved in [13], though we go one step further and derive the distribution of $\frac{X^T Y+1}{2}$. We follow a more geometrical argument and invite the reader to confer [13] for an alternative approach.

By the symmetry of the hypersphere, the distribution of $X^T Y$ is the same as the one of $X^T(1, 0 \dots, 0)$, which corresponds to rotating the reference frame. The cumulative distribution function then corresponds to the surface of the hyperspherical cap of angle $\cos^{-1}(X_1)$.
Using the formulas for the area of a spherical cap on S^D derived in [23], as well as the fact than $\sin^2(\cos^{-1}(x)) = 1 - x^2$ we directly obtain that for $X^T Y > 0$ (i.e. $\cos^{-1}(X_1) \leq \frac{\pi}{2}$), we have $1 - (X^T Y)^2 \sim \text{Beta}\left(\frac{D-1}{2}, \frac{1}{2}\right)$.
Since the density of the Beta distribution has reflectional symmetry, we see that $(X^T Y)^2 \sim \text{Beta}\left(\frac{1}{2}, \frac{D-1}{2}\right)$.

By substituting in $u = \frac{X^T Y+1}{2}$ if follows directly that

$$u \sim \text{Beta}\left(\frac{D-1}{2}, \frac{D-1}{2}\right), \tag{13}$$

concluding the proof. □

Lemma A.2. Considering an infinite amount of available negative samples, SimCLR and DCL's criteria lead to orthogonal embeddings as the dimension of embeddings M goes to infinity.

Proof. The proof hinges on Theorem 1 from [31], which states that as the number of negative samples goes to infinity, optimizing the repulsive force of the InfoNCE criterion leads to uniformly distributed embeddings on the M-hypersphere.

This uniform distribution allows us to leverage Lemma A.1 in saying that as the number of negative samples goes to infinity, for any pair of random embeddings X, Y, we have $\frac{X^T Y+1}{2} \sim \text{Beta}\left(\frac{M-1}{2}, \frac{M-1}{2}\right)$.
We can directly obtain the two following properties

$$\mathbb{E}\left[\frac{X^T Y + 1}{2}\right] = \frac{\frac{M-1}{2}}{\frac{M-1}{2} + \frac{M-1}{2}} = \frac{1}{2} \;\Rightarrow\; \mathbb{E}\left[X^T Y\right] = 0, \tag{14}$$

$$\text{Var}\left[\frac{X^T Y + 1}{2}\right] = \frac{\frac{M-1}{2} \times \frac{M-1}{2}}{\left(\frac{M-1}{2} + \frac{M-1}{2}\right)^2 \left(\frac{M-1}{2} + \frac{M-1}{2} + 1\right)} = \frac{1}{4M} \;\Rightarrow\; \text{Var}\left[X^T Y\right] = \frac{1}{M}. \tag{15}$$

This means that as the dimension of embeddings goes to infinity, the distribution of $X^T Y$ becomes a δ distribution, and so all dot products between negative pairs become 0, concluding the proof.

□

Proposition A.3. DCL and SimCLR are weakly-sample-contrastive.

Proof. Lemma A.2 tells us that with an infinite number of samples and embedding dimensions, SimCLR and DCL will learn orthogonal negative pair embeddings. This is the same minimizer as forL_{wc} since it leads to pairwise similarities of 0, and so SimCLR and DCL are weakly-sample-contrastive. □

Proposition A.4. SimCLR-abs/sq, DCL-sq/abs, as well as Spectral Contrastive Loss are sample-contrastive methods. Barlow Twins, VICReg and TCR are dimension-contrastive methods.

Proof. **DCL-sq/abs:** We first take a look at DCL-sq/abs's criteria. We consider that $\mathcal{K}$ is l2 normalized column-wise, i.e. embeddings are normalized. Let $f : \mathbb{R} \to \mathbb{R}^+$ be either defined as $f(x) = x^2$ for DCL-sq or as $f(x) = |x|$ for DCL-abs. We have

$$\mathcal{L}_{\text{DCL}} = \sum_{i=1}^{N} -\log\left(\frac{e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)/\tau}}{\sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}}\right) = \sum_{i=1}^{N} -\frac{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)}{\tau} + \log\left(\sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}\right). \tag{16}$$

The first part of this criterion is the invariance criterion and the second part is the LogSumExp(LSE) of embeddings' similarity. We know that this is a smooth approximation of the max operator with the following bounds:

$$\max\left(\{\forall j \neq i,\ f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)\}\right) \leq \tau \log\left(\sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}\right) \leq \max\left(\{\forall j \neq i,\ f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)\}\right) + \tau\log(N-1). \tag{17}$$

We can thus say that using either

$$\sum_{i=1}^{N} \log\left(\sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}\right) \quad \text{or} \quad \sum_{i=1}^{N} \max_{j\neq i} f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right), \tag{18}$$

as repulsive force will lead to the same result, a diagonal Gram matrix. Since this is the same goal as for our sample-contrastive criterion, DCL-sq and DCL-abs are sample-contrastive methods.

The link to L_c is more visible with the right term, which corresponds to only penalizing one value per row/column of the Gram matrix. While this is less effective than penalizing all of them at once, given sufficient training iterations it will converge to the same solution.

SimCLR-sq/abs: We now take a look at SimCLR-abs/sq's criteria. We consider that $\mathcal{K}$ is l2 normalized column-wise, i.e. embeddings are normalized. Let $f : \mathbb{R} \to \mathbb{R}^+$ be either defined as $f(x) = x^2$ for SimCLR-sq or as $f(x) = |x|$ for SimCLR-abs. We have

$$\mathcal{L}_{\text{SimCLR}} = \sum_{i=1}^{N} -\log\left(\frac{e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)/\tau}}{e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)/\tau} + \sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}}\right) \tag{19}$$

$$= \sum_{i=1}^{N} -\frac{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)}{\tau} + \log\left(e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right)/\tau} + \sum_{j\neq i} e^{f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)/\tau}\right). \tag{20}$$

Due to the presence of the positive pair in the repulsive force (right term), we cannot use the same reasoning with the max operator as for DCL-sq/abs which gave a clear intuition.

Nonetheless one can clearly see that to minimize this criterion, all the similarities between the negative pairs, i.e. $\forall i, \forall j \neq i,\ f\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)$, need to be minimized. As this will result in a diagonal Gram matrix, we can say that minimzing this criterion will also minimize our sample-contrastive one. We can thus conclude that SimCLR-sq and SimCLR-abs are sample-contrastive methods.

Spectral Constrastive Loss: We will now consider Spectral Constrastive Learning's criterion. We have

$$\mathcal{L}_{SCL} = -2\sum_{i=1}^{N} \mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i} + \sum_{j\neq i}\left(\mathcal{K}_{\cdot,i}^T\mathcal{K}_{\cdot,j}\right)^2 = -2\left(\sum_{i=1}^{N} \mathcal{K}_{\cdot,i}^T\mathcal{K}'_{\cdot,i}\right) + \|\mathcal{K}^T\mathcal{K} - \text{diag}(\mathcal{K}^T\mathcal{K})\|_F^2. \tag{21}$$

This means that Spectral Constrastive Loss also falls in the sample-contrastive category.

Barlow Twins: Looking at Barlow Twin's criterion we have

$$\mathcal{L}_{BT} = \sum_{j=1}^{M}\left(1 - (\mathcal{K}\mathcal{K}'^T)_{j,j}\right)^2 + \lambda \sum_{i,j,i\neq j}^{M} (\mathcal{K}\mathcal{K}'^T)_{j,i}^2 = \sum_{j=1}^{M}\left(1 - (\mathcal{K}\mathcal{K}'^T)_{j,j}\right)^2 + \lambda\|\mathcal{K}\mathcal{K}'^T - \text{diag}(\mathcal{K}\mathcal{K}'^T)\|_F^2. \tag{22}$$

Since the distribution of augmentations is the same for both views of the images, and the backbone is shared, taking a negative pair from $\mathcal{K}$ or $\mathcal{K}'$ is the same. Barlow Twins' criterion can then be rewritten as

$$\mathcal{L}_{BT} = \sum_{j=1}^{M}\left(1 - (\mathcal{K}\mathcal{K}'^T)_{j,j}\right)^2 + \lambda\|\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\|_F^2. \tag{23}$$

As such the right part of Barlow Twins' criterion is indeed the dimension-contrastive criterion, making Barlow Twins a dimension-contrastive method.

VICReg: VICReg's criterion is defined as

$$\mathcal{L}_{VICReg} = \lambda \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i} - \mathcal{K}'_{\cdot,i}\|_2^2 + \mu \left(v(\mathcal{K}) + v(\mathcal{K}')\right) + \nu \left(c(\mathcal{K}) + c(\mathcal{K}')\right). \tag{24}$$

Recall that c is a criterion that penalizes the off diagonal terms of the covariance matrix as follows:

$$c(\mathcal{K}) = \sum_{i \neq j} \text{Cov}(\mathcal{K})_{i,j}^2 = \|\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\|_F^2 = L_{nc}. \tag{25}$$

This means that VICReg is a dimension-contrastive method.

TCR: TCR's cost function is defined as

$$\mathcal{L}_{TCR} = -\frac{1}{2} \log\det\left(I + \alpha \text{Cov}(\mathcal{K})\right) = -\frac{1}{2} \log\det\left(I + \alpha \mathcal{K}\mathcal{K}^T\right) = -\frac{1}{2} \sum_i \log\left(1 + \alpha\sigma_i^2\right), \tag{26}$$

where σ_i is the i-th singular value of $\mathcal{K}$. As discussed in [24], this criterion leads to a diagonal covariance matrix, similarly to the non-contrastive criterion. We can thus say using either

$$-\frac{1}{2} \sum_i \log\left(1 + \alpha\sigma_i^2\right) \quad \text{or} \quad \|\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\|_F^2 \tag{27}$$

will lead to diagonal covariance matrices, or similarly null off-diagonal terms in the Covariance matrix. This means that TCR also falls in the category of dimension-contrastive methods. □

Theorem A.5. The sample-contrastive and dimension-contrastive criteria L_c and L_{nc} are equivalent up to row and column normalization of the embedding matrix $\mathcal{K}$. Consider a batch size of N and an embedding dimension of M. We have:

$$L_{nc} + \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 = L_c + \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4. \tag{28}$$

Proof. This proof is heavily inspired from the proof of Lemma 3.2 from [20] which provides a similar result for doubly stochastic matrices.
We have

$$\begin{aligned}
L_{nc} &= \|\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\|_F^2 && (29)\\
&= tr\left[\left(\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\right)^T \left(\mathcal{K}\mathcal{K}^T - \text{diag}(\mathcal{K}\mathcal{K}^T)\right)\right] && (30)\\
&= tr(\mathcal{K}\mathcal{K}^T\mathcal{K}\mathcal{K}^T) - 2tr(\mathcal{K}\mathcal{K}^T \,\text{diag}(\mathcal{K}\mathcal{K}^T)) + tr(\text{diag}(\mathcal{K}\mathcal{K}^T)\,\text{diag}(\mathcal{K}\mathcal{K}^T)) && (31)\\
&= tr(\mathcal{K}\mathcal{K}^T\mathcal{K}\mathcal{K}^T) - tr(\mathcal{K}\mathcal{K}^T \,\text{diag}(\mathcal{K}\mathcal{K}^T)) && (32)\\
&= tr(\mathcal{K}^T\mathcal{K}\mathcal{K}^T\mathcal{K}) - tr(\mathcal{K}\mathcal{K}^T \,\text{diag}(\mathcal{K}\mathcal{K}^T)). && (33)
\end{aligned}$$

Similarly for L_c, we obtain

$$\begin{aligned}
L_c &= \|\mathcal{K}^T\mathcal{K} - \text{diag}(\mathcal{K}^T\mathcal{K})\|_F^2 && (34)\\
&= tr(\mathcal{K}^T\mathcal{K}\mathcal{K}^T\mathcal{K}) - tr(\mathcal{K}^T\mathcal{K}\,\text{diag}(\mathcal{K}^T\mathcal{K})). && (35)
\end{aligned}$$

Since $\left(\mathcal{K}^T\mathcal{K}\right)_{i,i} = \|\mathcal{K}_{\cdot,i}\|_2^2$ we deduce that $tr(\mathcal{K}^T\mathcal{K}\,\text{diag}(\mathcal{K}^T\mathcal{K})) = \sum_{i=1}^{N} \|K_{\cdot,i}\|_2^4$. Similarly, we obtain that $tr(\mathcal{K}\mathcal{K}^T\,\text{diag}(\mathcal{K}\mathcal{K}^T)) = \sum_{j=1}^{M} \|K_{j,\cdot}\|_2^4$.
Plugging this back in, we finally deduce that

$$L_{nc} = L_c + \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4 - \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4, \tag{36}$$

concluding the proof.

□

Lemma A.6. If embeddings are normalized such that $\forall i,\ \|\mathcal{K}_{\cdot,i}\|_2 = a$ we have

$$\frac{N^2}{M} a^4 \leq \sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 \leq N^2 a^4. \tag{37}$$

Conversely, if dimensions are normalized such that $\forall j,\ \|\mathcal{K}_{j,\cdot}\|_2 = a$ we have

$$\frac{M^2}{N} a^4 \leq \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i}\|_2^4 \leq M^2 a^4. \tag{38}$$

Proof. We start with the first set of inequalities. Since $\forall i,\ \|\mathcal{K}_{i,\cdot}\|_2^2 \geq 0$ we have

$$\sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 \leq \left(\sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^2\right)^2 = \|\mathcal{K}\|_F^4 = N^2 a^4. \tag{39}$$

Which gives us our upper bound. For the lower bound, using the convexity of the function $f : x \to x^2$ we obtain

$$\frac{1}{M}\sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^4 \geq \left(\frac{1}{M}\sum_{j=1}^{M} \|\mathcal{K}_{j,\cdot}\|_2^2\right)^2 = \frac{N^2}{M^2} a^4. \tag{40}$$

Combining those two inequalities gives us the desired bounds.

For the second set of inequalities, we follow the same reasoning and use the fact that in this scenario $\|\mathcal{K}\|_F^2 = Ma^2$ giving us the aforementioned bounds and concluding the proof.

□

B Training procedure

For training, we follow common procedure and use a ResNet-50 backbone [18], with the LARS [33] optimizer. We use by default a base learning rate of 0.3 and compute the effective learning rate as $lr = base\_lr \times \frac{batch\_size}{256}$. We also use a momentum of 0.9 and weight decay of 10^{-6}. The learning rate follows a cosine annealing schedule after a 10-epoch linear warmup. We train for 100 epochs in all of our experiments.
For data augmentation, we follow the protocol of BYOL [14] which is as follows

Table 1: Image augmentation parameters, taken from [14].

Parameter	View 1	View 2
Random crop probability	1.0	1.0
Horizontal flip probability	0.5	0.5
Color jittering probability	0.8	0.8
Brightness adjustment max intensity	0.4	0.4
Contrast adjustment max intensity	0.4	0.4
Saturation adjustment max intensity	0.2	0.2
Hue adjustment max intensity	0.1	0.1
Grayscale probability	0.2	0.2
Gaussian blurring probability	1.0	0.1
Solarization probability.	0.0	0.2

Each experiment was run on 8 Nvidia V100 GPUs, with 32GB of memory each, and took around 24 hours to complete.

While this was our base experimental protocol, it was adapted for each method, mostly by changing method-specific hyperparameters as well as the learning rate, confer section H for the exact hyperparameters used for each experiment. The Pytorch pseudocode for VICReg-exp and VICReg-ctr is also available in section J.

C Online linear probe

As previously discussed, to evaluate our experiments, we relied on the use of a linear classifier that is trained jointly with our main network. This means that it is trained on suboptimal representations and stronger augmentations compared to what is typically done for linear evaluation. Even though these two approaches seem closely related, we are interested in finding how well they are correlated.

To do so, we trained a linear evaluation on VICReg and VICreg-exp with a projector architecture of $8192 - 8192 - d,\ d \in [256, 512, 1024, 2048, 8192]$ using the following protocol. We train the linear classifier on frozen representations for 100 epochs with a batch size of 1024 using the SGD optimizer with a base learning rate 0.25 (for VICReg) or 1.4 (for VICReg-exp), momentum 0.9, weight decay 10^{-6} and using a cosine annealing learning rate scheduler. We compute the learning rate as $lr = base\_lr \times \frac{batch\_size}{256}$. For augmentations, we follow standard procedure and use random cropping with a scale between 0.08 and 1 with an image size of 224×224 and horizontal flip with a probability 0.5 during training. For evaluation, we do a center crop.

Table 2: Relationship in performance between the online linear probe and the offline linear classifier. We used VICReg and an expander with architecture $8192 - 8192 - d$.

Embedding dimension	256	512	1024	2048	8192
Online top-1	65.01	66.72	68.06	68.06	68.13
Offline top-1	65.11	66.64	67.96	68.00	68.02

Table 3: Relationship in performance between the online linear probe and the offline linear classifier. We used VICReg-exp and an expander with architecture $8192 - 8192 - d$.

Embedding dimension	256	512	1024	2048	8192
Online top-1	65.24	66.71	67.86	68.00	67.93
Offline top-1	65.30	66.58	67.83	67.89	68.18

As we can see in table 2 and 3, the performance achieved by the offline classifier is extremely close to the performance of the online classifier. While the online classifier cost in compute is negligible, the linear evaluation is almost as long as the pretraining due to data loading bottlenecks and it requires a significant amount of learning rate tuning. This makes this online classifier a very appealing alternative since it demonstrates very correlated performances for a fraction of the computing cost.

Training a linear regression on those two sets of evaluations gives a model with a slope of 0.97, an intercept of 2.1, and an R^2 of 1.0. It is worth noting that since most values are close to 68, the fitting of linear regression on this data is sensitive to noise. Nonetheless, the low intercept, as well as the closeness of the slope to 1, confirm the negligible gap between the two evaluation methods that we previously intuited.

D Impact of the similarity measure on SimCLR

While SimCLR uses cosine similarity to push away negative pairs, we will look at what happens when we use the square or absolute value of cosine similarities, as in SimCLR-sq or SimCLR-abs.

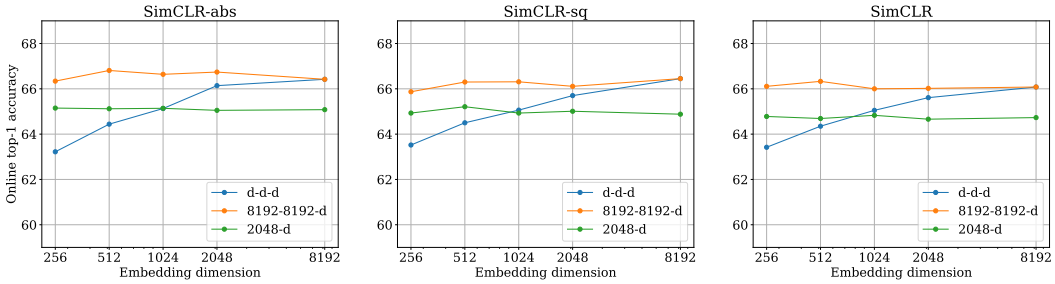


Figure 3: Influence of using squared or absolute values of the cosine similarities for VICReg, with different projector architectures.

As we can see in figure 3, the use of the squared or absolute values of the similarities did not impact the performance of image classification, it even improved with a large projector when using the absolute values.

As we can see in figure 4, for all three methods we obtain a distribution of cosine similarities that is centered at 0, but they all have very different standard deviations. The main culprit of this difference is dimensional collapse, as studied extensively in [19]. As we can see in figure 5, the three methods show different levels of collapse. While SimCLR-abs appears to have an almost full rank embedding matrix, we can see some collapse at around 256 dimensions for SimCLR, and 64 for SimCLR-sq. Per the proof of Lemma A.2, we know that with a perfect optimization of SimCLR's criterion, we should observe a variance of $1/D$ for the cosine similarities, if we have D-dimensional embeddings. However this is not the ambient dimension but the embeddings dimension, and so when combining this result with the dimensional collapse, we clearly see that SimCLR-abs should have less variance as it has the least amount of collapse, and SimCLR-sq the highest variance as it has the most amount of collapse. Since this is what we observe in practice, these results are coherent with the three methods producing similar cosine similarities distributions, albeit with different standard deviations depending on the amount of dimensional collapse.

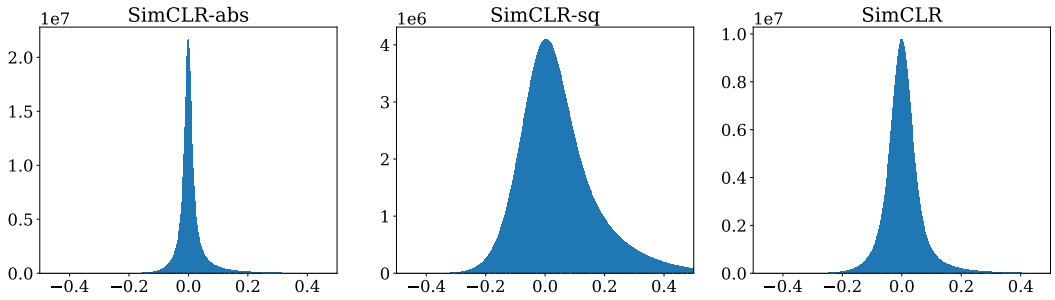
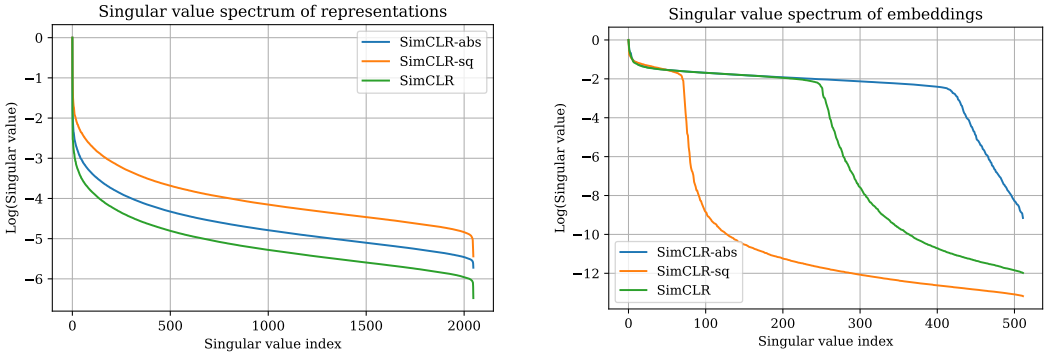


Figure 4: Histogram of cosine similarities for negative pairs in SimCLR-abs, SimCLR-sq and SimCLR.



(a) Representations

(b) Embeddings

Figure 5: Singular value distribution of the embeddings and representations computed on the training set of ImageNet for SimCLR, SimCLR-abs and SimCLR-sq. All methods use 512 dimensional embeddings.

E Row and column norms interplay

While we provided bounds that apply to any matrix in lemma 3.4, in practice embedding matrices have a particular structure and one can wonder where the norms are in between the relatively distant bounds.
To study this we took 1024 images from ImageNet, computed the corresponding embedding matrices, and then l2-normalized the rows or columns.

Table 4: The empirical interplay between embedding matrix norms under row- or column-wise l2-normalization for different methods and projector architectures. We abbreviate thousands with k and millions with M. The experiment "Random" indicates a randomly initialized network.

Experiment	Projector	Colum normalization			Row normalization		
		$\frac{N^2}{M}$	$\sum \|\mathcal{K}_{j,\cdot}\|_2^4$	N^2	$\frac{M^2}{N}$	$\sum \|\mathcal{K}_{\cdot,i}\|_2^4$	M^2
VICReg	$8192-8192-8192$	128	128.19	1M	65k	83k	67M
VICReg-exp	$8192-8192-8192$	128	128.26	1M	65k	95k	67M
VICReg-ctr	$8192-8192-512$	2048	2078	1M	256	287	262k
SimCLR	$8192-8192-512$	2048	2061	1M	256	433.54	262k
	$8192-8192-8192$	128	129.43	1M	65k	113k	67M
Random	$8192-8192-8192$	128	361.34	1M	65k	75k	67M

As we can see in table 4, for every method, in any expansion or projection scenario, we are always close to the lower bound, deviating by a factor of 3 at most. This is significantly smaller than the factors N or M in lemma 3.4 which are tight when making no assumptions on the embedding matrix $\mathcal{K}$. As previously discussed these extreme cases consist respectively of a constant matrix and one with only one non-zero element per

row/column. It is logical that the embedding matrices that we have in practice are closer to a constant matrix, with a uniform spread of information, even though they still present some sparsity.
As such, for all practical concerns the bounds are much closer in practice than they theoretically are. This means that the sample-contrastive and dimension-contrastive criteria will also be closer in practice.

F Embedding vs dimension normalization in VICReg-ctr

While we normalized the embeddings with VICReg-ctr, as is usually done in contrastive methods, a legitimate question is if this choice of normalization plays a role in the overall performance or behavior of the method. To this effect, we implemented a version of VICReg-ctr which normalized the dimensions, as is done for VICReg and VICReg-exp. Its loss function is thus

$$\mathcal{L}_{VICReg-ctr'} = \lambda \sum_{i=1}^{N} \|\mathcal{K}_{\cdot,i} - \mathcal{K}'_{\cdot,i}\|_2^2 + \mu \left(v(\mathcal{K}) + v(\mathcal{K}')\right) + \nu \left(c_{exp}(\mathcal{K}^T) + c_{exp}(\mathcal{K}'^T)\right)$$

This has the benefit of completely isolating the contrastiveness of the criterion when compared to VICReg-exp, though it comes with a significant drawback. This change in normalization forced us to remove the mixed precision when training, as it is very unstable due to the high embedding norms. While this only has practical implications, the significant increase in memory consumption and training time makes this approach less appealing than its counterpart, which normalized the embeddings.

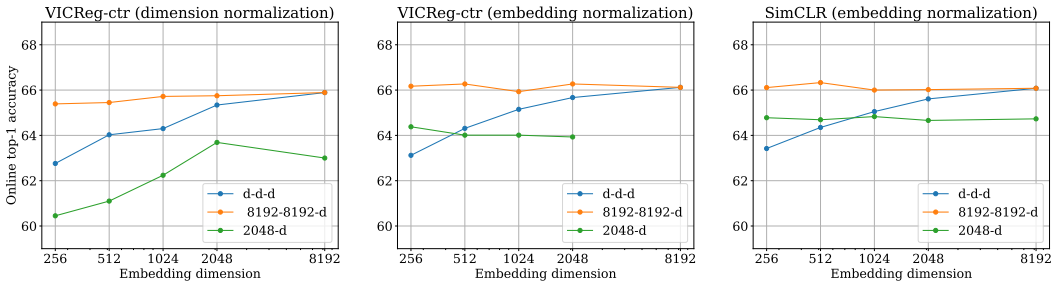


Figure 6: Behavior of VICReg-ctr with different normalization with respect to embedding dimension when changing the projector's architecture, compared to SimCLR.

As we can see in figure 6, we observe a similar behavior when normalizing in either direction for VICReg-exp with a projector following the architectures $8192 - 8192 - d$ or $d - d - d$, although the performance is slightly lower than before. We also see a small decrease in performance as the embedding dimension decreases when normalizing the dimensions. However, when using a projector with architecture $2048 - d$, the performance is significantly lower when normalizing the dimensions, and shows a significant decrease in performance as the embedding dimension decreases, which was not the case when normalizing the embeddings or for SimCLR. While this suggests that it is better to normalize the embeddings in contrastive settings, in both cases we can see better robustness to embedding dimensions compared to VICReg and VICReg-exp.

G Impact of the projector capacity

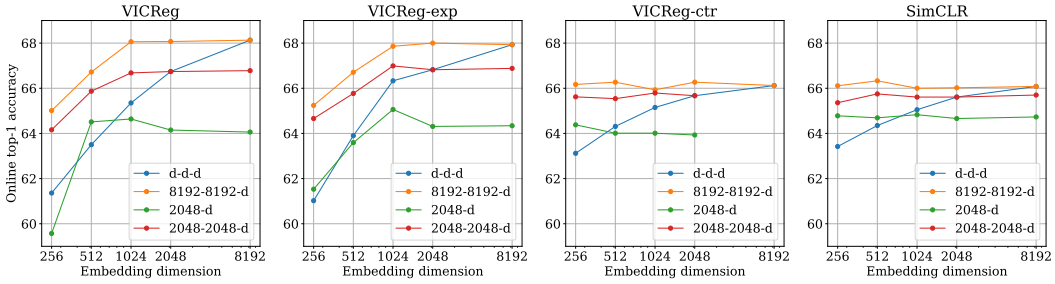


Figure 7: Online performance on ImageNet for VICReg, VICReg-exp, VICReg-ctr, and SimCLR with respect to embedding dimensions when changing the projector's architecture.

As discussed in section 5, the design of the projector plays a significant role in downstream performance. In figure 7, we also overlay the results for a projector with architecture $2048 - 2048 - d$ on top of the previously discussed ones. Such a projector offers similar behavior as an $8192 - 8192 - d$ one, but with a bit lower performance. The drop in performance is especially noticeable in dimension-contrastive methods.

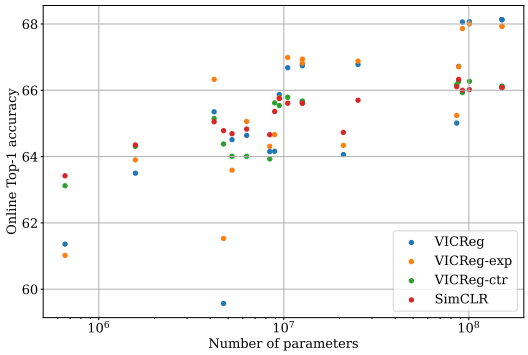


Figure 8: Online performance on ImageNet for VICReg, VICReg-exp, VICReg-ctr, and SimCLR with respect to the number of parameters in their projector.

As we can see in figure 8, if we take a look at the performance with respect to the number of parameters of the projector we can see a clear trend that indicates that performance is increased when increasing the number of parameters of the projector. This conclusion holds for all methods though there are some scenarios that are clear outliers. For example, for VICReg and VICReg exp we can see that with a $2048 - 256$ projector, the performance is significantly lower than expected.

While it would be interesting to see if this increase in performance saturates at some point, our largest projectors already have 151 million parameters. Increasing it further quickly starts to become impractical due to memory constraints during training, and as such, we leave this study to future work.

Another aspect worth mentioning is that the increase in performance when increasing the number of parameters is not automatic. For example for VICReg, the scenario $2048 - 2048 - 1024$ achieves 66.68% top-1 for 10 million parameters, but the scenario $8192 - 8192 - 256$ only achieves 65.01% even though it has 86 million parameters. This drastic difference suggests that some care must be taken when designing the projector and that even though the number of parameters is important, the architecture in itself also is.

H Complete performance and hyperparameter tables

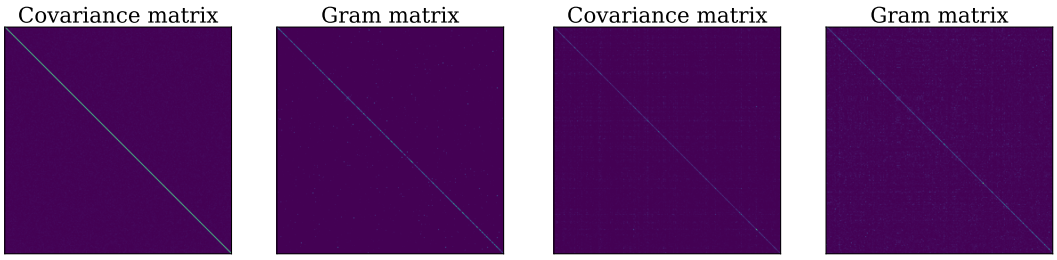
Table 5: Top-1 accuracy on ImageNet using the online linear classifier, including all performances for figures 2 and 7.

Experiment	Projector	256	512	1024	2048	8192
VICReg	$d-d-d$	61.36	63.50	65.35	66.74	68.13
	$8192-8192-d$	65.01	66.72	68.06	68.07	68.13
	$2048-d$	59.57	64.51	64.64	64.15	64.06
	$2048-2048-d$	64.16	65.87	66.68	66.74	66.78
VICReg-exp	$d-d-d$	61.02	63.90	66.33	66.82	67.93
	$8192-8192-d$	65.24	66.71	67.86	68.00	67.93
	$2048-d$	61.53	63.59	65.06	64.31	64.34
	$2048-2048-d$	64.66	65.77	66.99	66.82	66.88
VICReg-ctr	$d-d-d$	63.12	64.31	65.15	65.67	66.12
	$8192-8192-d$	66.17	66.27	65.93	66.27	66.12
	$2048-d$	64.38	64.01	64.01	63.93	N/A
	$2048-2048-d$	65.62	65.54	65.79	65.67	N/A
SimCLR	$d-d-d$	63.42	64.35	65.05	65.61	66.08
	$8192-8192-d$	66.11	66.33	66.00	66.02	66.08
	$2048-d$	64.78	64.69	64.83	64.66	64.73
	$2048-2048-d$	65.36	65.75	65.61	65.61	65.70

Table 6: Hyperparameters used for the results in table 5. Sim., Var. and Cov. indicate the weights of the criteria in VICReg and its variations. τ indicates the temperature used for LogSumExp based methods. The hyperparameters for VICReg and SimCLR are usable with the official implementations. For VICReg-exp and VICReg-ctr they are compatible with the pseudocode in section J.

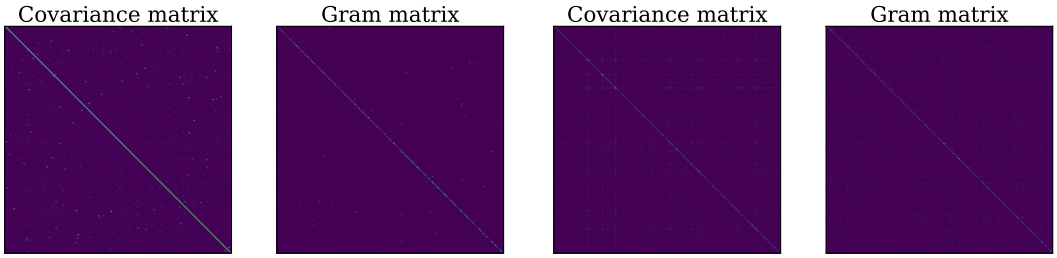
Experiment	Projector	Batch size	base lr	VICReg			τ
				Sim.	Var.	Cov.	
VICReg	$d=256$	1024	0.3	25	25	4	
	$d=512$	1024	0.3	25	25	2	
	$d=1024$	1024	0.3	25	25	2	
	$d=2048$	1024	0.3	25	25	2	
	$d=8192$	1024	0.3	25	25	0.5	
VICReg-exp	$d \neq 8192$	1024	0.5	1	1	2	0.1
	$d=8192$	1024	0.8	1	1	2	0.1
VICReg-ctr	All	1024	0.5	1	1	1	0.1
SimCLR	All	2048	0.6				0.1

I Larger covariance and Gram matrices



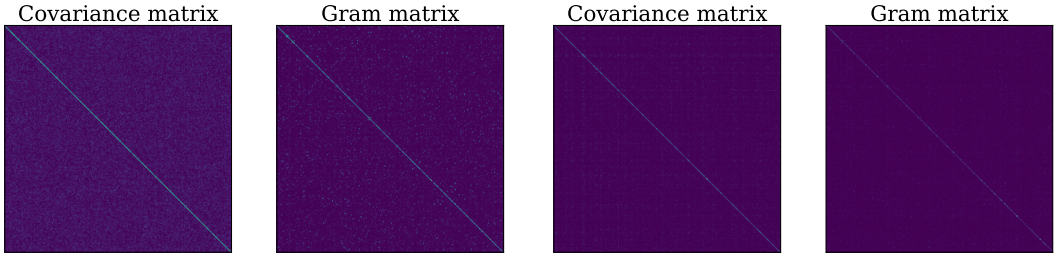
(i) On the embeddings (ii) On the representations

(a) VICReg



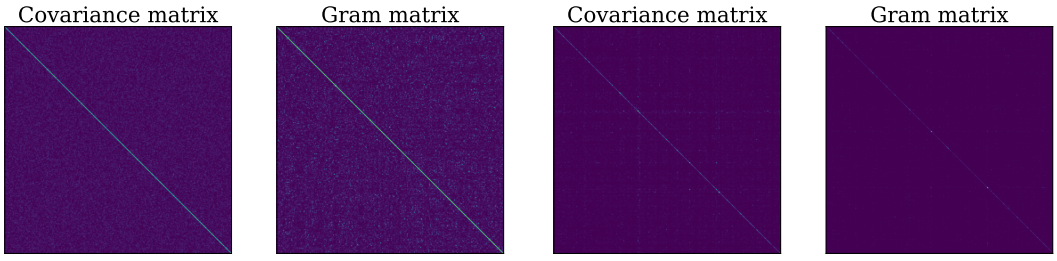
(i) On the embeddings (ii) On the representations

(b) VICReg-exp



(i) On the embeddings (ii) On the representations

(c) VICReg-ctr



(i) On the embeddings (ii) On the representations

(d) SimCLR

Figure 9: Covariance and similarity matrices on a random part of the ImageNet training set, using VICReg, VICReg-exp, VICReg-ctr, and SimCLR pretrained on ImageNet for 100 epochs. The covariance matrix is limited to the first 256 dimensions, while the Gram matrix is limited to the first 256 samples. In all cases, we used a projector with an output dimension of 2048, the same as the representation dimension.

J VICReg variations pseudocode

Algorithm 1: VICReg-exp pytorch pseudocode.

```
# f: encoder network,p: projector network, lambda, mu, nu:
    coefficients of the invariance, variance and covariance losses, N:
     batch size, D: dimension of the representations, tau: temperature
# mse_loss: Mean square error loss function, relu: ReLU activation
    function, cut_out_diag: remove the diagonal of a matrix,

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    x_a, x_b = augment(x)

    # compute embeddings
    k_a = p(f(x_a)) # N x D
    k_b = p(f(x_b)) # N x D

    # invariance loss
    sim_loss = mse_loss(k_a, k_b)

    # variance loss
    std_k_a = torch.sqrt(k_a.var(dim=0) + 1e-04)
    std_k_b = torch.sqrt(k_b.var(dim=0) + 1e-04)
    std_loss = torch.mean(relu(1 - std_k_a))/2 + torch.mean(relu(1 -
        std_k_b))/2

    # covariance loss
    k_a = k_a - k_a.mean(dim=0)
    k_b = k_b - k_b.mean(dim=0)
    cov_k_a = (k_a.T @ k_a) / (N - 1)
    cov_k_b = (k_b.T @ k_b) / (N - 1)
    cov_loss = torch.logsumexp(cut_out_diag(cov_k_a/tau),1).mean()/2 +
              torch.logsumexp(cut_out_diag(cov_k_b/tau),1).mean()/2

    # loss
    loss = lambda * sim_loss + mu * std_loss + nu * cov_loss

    # optimization step
    loss.backward()
    optimizer.step()
```

Algorithm 2: VICReg-ctr pytorch pseudocode.

```
# f: encoder network,p: projector network, lambda, mu, nu:
    coefficients of the invariance, variance and covariance losses, N:
     batch size, D: dimension of the representations, tau: temperature
# mse_loss: Mean square error loss function, relu: ReLU activation
    function, cut_out_diag: remove the diagonal of a matrix,

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    x_a, x_b = augment(x)

    # compute embeddings
    k_a = p(f(x_a)) # N x D
    k_b = p(f(x_b)) # N x D

    # invariance loss
    sim_loss = mse_loss(k_a, k_b)

    #Make the method contrastive
    k_a = k_a.T
    k_b = k_b.T

    # variance loss
    std_k_a = torch.sqrt(k_a.var(dim=0) + 1e-04)
    std_k_b = torch.sqrt(k_b.var(dim=0) + 1e-04)
    std_loss = torch.mean(relu(1 - std_k_a))/2 + torch.mean(relu(1 -
        std_k_b))/2

    # covariance loss
    k_a = k_a - k_a.mean(dim=0)
    k_b = k_b - k_b.mean(dim=0)
    cov_k_a = (k_a.T @ k_a) / (N - 1)
    cov_k_b = (k_b.T @ k_b) / (N - 1)
    cov_loss = torch.logsumexp(cut_out_diag(cov_k_a/tau),1).mean()/2 +
              torch.logsumexp(cut_out_diag(cov_k_b/tau),1).mean()/2

    # loss
    loss = lambda * sim_loss + mu * std_loss + nu * cov_loss

    # optimization step
    loss.backward()
    optimizer.step()
```