VARIANCE COVARIANCE REGULARIZATION ENFORCES PAIRWISE INDEPENDENCE IN SELF-SUPERVISED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-Supervised Learning (SSL) methods such as VICReg, Barlow Twins or W-MSE avoid collapse of their joint embedding architectures by constraining or regularizing the covariance matrix of their projector's output. This study highlights important properties of such strategy, which we coin Variance-Covariance regularization (VCReg). More precisely, we show that *VCReg enforces pairwise independence between the features of the learned representation*. This result emerges by bridging VCReg applied on the projector's output to kernel independence criteria applied on the projector's input. This provides the first theoretical motivations and explanations of VCReg. We empirically validate our findings where (i) we put in evidence which projector's characteristics favor pairwise independence, (ii) we use these findings to obtain nontrivial performance gains for VICReg, (iii) we demonstrate that the scope of VCReg goes beyond SSL by using it to solve Independent Component Analysis. We hope that our findings will support the adoption of VCReg in SSL and beyond.

1 INTRODUCTION

Self-Supervised Learning (SSL) via joint embedding architectures has risen to learn visual representations outperforming their supervised counterpart. This paradigm typically enforces similar embeddings for two augmented versions of the same sample, thus allowing an encoder f to learn a representation for a given modality without labels. Importantly, f could solve the learning task by predicting the same embedding for every input, a failure mode known as collapse. To avoid this, various mechanisms have been proposed hence the diversity of SSL methods (*e.g.*, Grill et al. (2020); Caron et al. (2020; 2021); Chen & He (2021)). Most of these are a composition $g \circ f$ of the encoder with a projector neural network g. Only f is retained after training, in opposition to supervised training that never introduces g. The projector was proposed by Chen et al. (2020) and significantly improved the quality of the learned representation in terms of test accuracy on ImageNet and other downstream tasks. Although various works (Appalaraju et al., 2020; Cosentino et al., 2022; Bordes et al., 2022) provide empirical knowledge on the projector, its exact role remains poorly understood.

Variance-Covariance Regularization (VCReg) was introduced in recent methods for SSL with joint embeddings (Bardes et al., 2022; Zbontar et al., 2021; Ermolov et al., 2021) to cope with collapse by constraining or regularizing the covariance or cross-correlation of the projector g output to be identity. Bardes et al. (2022); Zbontar et al. (2021) observe that wider projectors further improve the representation learned by the encoder f and the former alludes to the possibility that decorrelating a non-linear projection of the representation improves the quality of the learned representation. However, neither work further study these intriguing phenomena.

This study sheds a new light on the role of the projector by demonstrating that *VC regularization of the projector's output precisely enforces pairwise independence for the components of the projector's input i.e. the encoder's output*, and connects this property to projector's parameters such as width and depth. Fully or partially pairwise independent representations are generally sought for, *e.g* to disentangle factors of variations (Li et al., 2019; Träuble et al., 2021). Our experimental analysis suggests that different levels of pairwise independence of the features in the representation emerge from a variety of SSL criteria along with mutual independence. However, as opposed to other

frameworks, VCReg allows for theoretical study and explicit control of the learned independence amount. We prove and experimentally validate this property for random projectors, study how it emerges in learned projectors, and use our results to obtain new and significant performance gains on VICReg over Bardes et al. (2022). We then ablate the SSL context and lean on our findings to show that VCReg is sufficient to solve Independent Component Analysis (ICA). Beyond providing a novel theoretical understanding of VCReg, we believe that this work also leads to a better understanding of the projector in SSL methods. The scope of VCReg is not limited to SSL: our experiments on ICA open the way to other applications where some degree of independence is needed.

2 Background

2.1 Measuring Statistical Independence Using Kernel Methods

Measuring the independence between two set of realizations $\{\boldsymbol{X}_1^1, \dots, \boldsymbol{X}_1^N\}, \{\boldsymbol{X}_2^1, \dots, \boldsymbol{X}_2^N\}, \boldsymbol{X}_1^i \in \mathbb{R}^M, \boldsymbol{X}_2^i \in \mathbb{R}^M$ is a fundamental task that has a long history in statistics e.g. through the Mutual Information (MI) of the two random variables X_1 and X_2 from which those two sets are independently drawn from (Cover, 1999). Those variables are said independent if the realization of one does not affect the probability distribution of the other. Computing the MI in practice is known to be challenging (Goebel et al., 2005), which has led to considerable interest in using alternative criteria e.g. based on functions in Reproducing Kernel Hilbert Spaces (RKHS) (Bach & Jordan, 2002), a special case of what is known as functional covariance or correlation (Rényi, 1959). It consists in computing those statistics after nonlinear transformation (Leurgans et al., 1993) as in

$$\sup_{f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2} \mathrm{Corr}\langle f_1(X_1), f_2(X_2) \rangle, \tag{1}$$

where f_1, f_2 are constrained to lie within some designed functional space, and the Cov can be used instead of the Corr. If Eq. (1) is small enough, then X_1 and X_2 are independent in regard to the functional spaces $\mathcal{F}_1, \mathcal{F}_2$. For example, if $\mathcal{F}_1$ and $\mathcal{F}_2$ are unit balls in their respective vector spaces, then Eq. (1) is just the norm of the usual correlation/covariance operator (Mourier, 1953) which would be enough for independence under joint Gaussian distributions (Melnick & Tenenbein, 1982).

HSIC and pairwise independence. More recently, Gretton et al. (2005a) introduced a pairwise independence criterion known as the Hilbert-Schmidt Independence Criterion (HSIC) which can be estimated given empirical samples $\boldsymbol{X}_1$ and $\boldsymbol{X}_2 \in \mathbb{R}^{N \times M}$ as

$$\mathrm{HSIC}(\boldsymbol{X}_1, \boldsymbol{X}_2) := \widehat{\mathrm{HSIC}}(X_1, X_2) = \frac{1}{(N-1)^2} \mathrm{Tr}(\boldsymbol{K}_1 \boldsymbol{H} \boldsymbol{K}_2 \boldsymbol{H}), \tag{2}$$

with $\boldsymbol{H}$ the centering matrix $\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T \frac{1}{N}$, $(\boldsymbol{K}_1)_{i,j} = k_1(\boldsymbol{X}_1^i, \boldsymbol{X}_1^j)$ and $(\boldsymbol{K}_2)_{i,j} = k_2(\boldsymbol{X}_2^i, \boldsymbol{X}_2^j)$ the two kernel matrices of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ respectively, and k_1, k_2 of $\mathcal{F}_1, \mathcal{F}_2$ universal kernels such as the Gaussian kernel (see Steinwart (2001); Micchelli et al. (2006) for other examples). Crucially, since

$$\mathrm{HSIC}(X_1, X_2) \geq \sup_{f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2} \mathrm{Cov}\langle f_1(X_1), g(X_2) \rangle, \tag{3}$$

HSIC can be used to test for (pairwise) independence as formalized below.

Theorem 1 (Thm. 4 from (Gretton et al., 2005a))**.** $\mathrm{HSIC}(X_1, X_2) = 0$ *if and only if* X_1 *and* X_2 *are independent.*

Gretton et al. (2005a) also provide a statistical test for pairwise independence based on HSIC. Further quantities such as upper bounds on the MI can be found in a similar way, e.g. see Thm. 16 in Gretton et al. (2005b). In our experiments, we will rely on HSIC under the Gaussian kernel scaled by the median of the distribution of pairwise euclidean distances between samples.

dHSIC and mutual independence. Mutual independence of a set of D random variables $X_1, \dots, X_D$ is a stronger property than independence between all pairs of random variables in the set. To evaluate it, Pfister et al. (2018) introduce dHSIC, a multivariate extension of HSIC. In short, dHSIC measures the distance between mean embeddings μ under a RKHS $\mathcal{F}$ (Smola et al., 2007) of the product of distributions and the joint distribution:

$$\mathrm{dHSIC}(X_1, \dots, X_D) := \|\mu(\mathbb{P}^{X_1} \otimes \dots \otimes \mathbb{P}^{X_D}) - \mu(\mathbb{P}^{X_1, \dots, X_D})\|_{\mathcal{F}}. \tag{4}$$

Similarly to HSIC, Pfister et al. (2018) establish the equivalence between null dHSIC and mutual independence along with a statistical test. We provide an estimator of dHSIC given empirical samples $\boldsymbol{X}_1, \dots, \boldsymbol{X}_D$ in $\mathbb{R}^{N \times M}$ each as well as implementations of HSIC and dHSIC in Appendix D.

Complexities. In what follows, we consider scalar random variables ($M = 1$). The subsequent complexities of HSIC and dHSIC are $\mathcal{O}(N^2)$ and $\mathcal{O}(DN^2)$ respectively. Testing pairwise independence between all features in a D-set with HSIC is $\mathcal{O}(D^2N^2)$ while dHSIC requires $N \geq 2D$ (Pfister et al., 2018). Since competitive visual representations typically have $D = 2048$, we resort to surrogates to estimate independence of the representations in practice. These surrogates are detailed in Section 5.

2.2 Variance-Covariance Regularization in Self-Supervised Learning

SSL with joint embeddings learns visual representations by producing two different augmented views of an input batch of images $\boldsymbol{S} \in \mathbb{R}^{N \times W \times H}$, denoted by $\boldsymbol{S}_{\text{left}}$ and $\boldsymbol{S}_{\text{right}}$. Each view is fed to an encoder f, typically a ResNet50 (He et al., 2016), producing representations $\boldsymbol{X}_{\text{left}}$ and $\boldsymbol{X}_{\text{right}} \in \mathbb{R}^{N \times D}$ which are passed through a projector g to output embeddings $\boldsymbol{Z}_{\text{left}}$ and $\boldsymbol{Z}_{\text{right}} \in \mathbb{R}^{N \times P}$. Finally, an invariance term encouraging $\boldsymbol{Z}_{\text{left}}$ and $\boldsymbol{Z}_{\text{right}}$ to be similar is applied. While most SSL methods require architectural or training strategies to avoid collapse (Grill et al., 2020; He et al., 2021), Bardes et al. (2022) and Zbontar et al. (2021) only require to modify the loss. After training, only f is retained to be used in downstream tasks. We will denote the $(2N, P)$ matrix $\boldsymbol{Z} \triangleq [\boldsymbol{Z}_{\text{left}}^T, \boldsymbol{Z}_{\text{right}}^T]^T$.

VICReg. In Bardes et al. (2022), an anti-collapse term $\mathcal{L}_{\text{VC}}$, which we coin VC regularization (VCReg), is added to an invariance loss to form $\mathcal{L}_{\text{VIC}}$:

$$\mathcal{L}_{\text{VC}} = \sum_{k=1}^{P} \max\left(0, 1 - \sqrt{\text{Cov}(\boldsymbol{Z})_{k,k}}\right) + \alpha \sum_{j=1, j \neq k}^{P} \text{Cov}(\boldsymbol{Z})_{k,j}^2, \tag{5}$$

$$\mathcal{L}_{\text{VIC}} = \frac{1}{N} \sum_{n=1}^{N} \|(\boldsymbol{Z}_{\text{left}})_{n,\cdot} - (\boldsymbol{Z}_{\text{right}})_{n,\cdot}\|_2^2 + \mathcal{L}_{\text{VC}}, \tag{6}$$

The leftmost term in $\mathcal{L}_{\text{VC}}$ corresponds to regularizing the variance of each feature in $\boldsymbol{Z}$ to be unit, while the second term seeks to minimize the covariance between each pair of features in $\boldsymbol{Z}$. In Bardes et al. (2022), the weight of each term in $\mathcal{L}_{\text{VIC}}$ can be tuned. Since the authors find that best results are obtained with equal weights for the invariance and the variance terms, we only vary α. Although this work focus on VCReg as formulated in VICReg, we provide background on Barlow Twins and W-MSE in the Appendix A, while next section establishes the similarity between VICReg, Barlow Twins and W-MSE as VCReg optimizers.

3 VC Regularization of SSL Projector's Output Enforces Pairwise Independent Features at the Encoder's Output

In this section, we demonstrate that for random Multi-Layer Perceptrons (MLPs) projectors, minimizing VC of the projector's output amounts to minimizing HSIC —a measure of pairwise dependence (see Section 2.1)— between all pairs of variables in the learned representation, *i.e* the projector's input. We then justify how this reasoning extends to learned projectors. Our claims are experimentally verified in Section 5.

Notations. In our setting, $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ is the D dimensional output of the encoder f for a batch of size N. $\boldsymbol{X}$ is then fed into the projector g to form embeddings $\boldsymbol{Z} = g(\boldsymbol{X}) = [g(\boldsymbol{X})_1, \dots, g(\boldsymbol{X})_P] \in \mathbb{R}^{N \times P}$ on which SSL criteria are generally applied. The acronym MLP will refer to projectors typically used in SSL: neural network with three layers of same width unless stated otherwise.

Our proof strategy consists in proving that VCReg of each MLP layer (Linear + Batch Normalization (BN) + ReLU) output enforces pairwise independence of its input, before composing these results. We first study nonlinear elementwise projectors $g : \mathbb{R} \mapsto \mathbb{R}^L$, which belong to the wider class of DeepSets (Zaheer et al., 2017), and of which BN followed by ReLU can be seen as an instance. We will denote the mapping of such projectors as $\boldsymbol{Z} = g(\boldsymbol{X}) = [g(\boldsymbol{X}_{:,1}), \dots, g(\boldsymbol{X}_{:,P})]$.

Lemma 1 (Nonlinear elementwise projectors minimize HSIC of their input)**.** *Let $g : \mathbb{R} \mapsto \mathbb{R}^L$ be a nonlinear elementwise projector; then, minimizing the covariance of $\boldsymbol{Z}$ with respect to the encoder f amounts to minimizing HSIC on all feature pairs in $\boldsymbol{X}$ with kernels $K_i = g((\boldsymbol{X})_{:,i})g((\boldsymbol{X})_{:,i})^T$.*

Proof. Let us consider the N values of the ith data feature $(\boldsymbol{X}_{:,i})$ as realizations of a random variable. Recalling Eq. (2), independence of two random variables can be estimated via HSIC. Considering the arbitrarily complicated network g and $\boldsymbol{Z} = [g(\boldsymbol{X}_{:,1}), \dots, g(\boldsymbol{X}_{:,D})] \in \mathbb{R}^{N \times DL}$, we have:

$$\begin{aligned}
\text{HSIC}((\boldsymbol{X})_{:,i}, (\boldsymbol{X})_{:,j}) &= \frac{1}{(N-1)^2} \operatorname{Tr}(g((\boldsymbol{X})_{:,i})g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j})g((\boldsymbol{X})_{:,j})^T \boldsymbol{H}) \\
&= \frac{1}{(N-1)^2} \left\| g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j}) \right\|_F^2 \\
&= \left\| \text{Cov}\left(g((\boldsymbol{X})_{:,i}), g((\boldsymbol{X})_{:,j})\right) \right\|_F^2 \\
&= \left\| \text{Cov}\left(\boldsymbol{Z}\right)_{1+iL:1+(i+1)L, 1+jl:1+(j+1)L} \right\|_F^2 \\
\implies \sum_{i \neq j} \text{HSIC}((\boldsymbol{X})_{:,i}, (\boldsymbol{X})_{:,j}) &= \left\| \text{Cov}\left(\boldsymbol{Z}\right) \odot \left((1 - \boldsymbol{I}_D) \otimes \mathbf{1}_L \mathbf{1}_L^T\right) \right\|_F^2,
\end{aligned}$$

and, in the case $L = 1$ we have $\mathbf{1}_L \mathbf{1}_L^T = 1$ leading to $\sum_{i \neq j} \text{HSIC}((\boldsymbol{X})_{:,i}, (\boldsymbol{X})_{:,j}) = \sum_{i \neq j} \text{Cov}\left(\boldsymbol{Z}\right)_{i,j}^2$, concluding the proof. □

To rigorously obtain independence, the K_i's must be universal kernels. To satisfy this assumption, we may randomize Batch Normalization by drawing the mean and variance from some distribution. Combining this operation with a ReLU, we obtain random elementwise nonlinearities approaching random features (Rahimi & Recht, 2007) of a universal kernel (Sun et al., 2018). Increasing L improves the approximation of such kernel (Chen & Phillips, 2017) i.e. the larger L, the better approximation of HSIC will VICReg's covariance term be; see Appendix D for the randomized Batch Normalization implementation, Figure 7 for its justification, and Section 5 for empirical validation.

Remark 2 (Necessity of variance regularization)**.** *Although the variance regularization term does not explicitly appear when minimizing HSIC on all pairs, it is necessary when optimizing $\boldsymbol{X}$ to prevent the degenerate solution of $\boldsymbol{X}$ being a constant, a common collapse mode of SSL.*

Lemma 2 (Random linear projectors minimize HSIC of their input)**.** *Let g be a random linear projector and $\boldsymbol{X}$ has same variance for each column. Then, minimizing the covariance of $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}$ with respect to the encoder f amounts to minimizing HSIC with a linear kernel for each pair of variables in $\boldsymbol{X}$. (Proof in Appendix B.1.)*

Since the corresponding kernel in lemma 2 is linear, only decorrelation can be achieved for such projector's input. This however differs from PCA as we optimize over $\boldsymbol{X}$ and not g's parameters ($\boldsymbol{W}$). Proving lemma 2 requires a projector with orthogonal weights, i.e. $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$, which gets more and more accurate with random weights $\mathcal{U}(1/\sqrt{D}, 1/\sqrt{D})$ as P (the output dimension of g) increases. This follows since the central limit theorem states that the dot-product between two P-dimensional weight vectors tends to 0 with rate $\mathcal{O}(1/\sqrt{P})$. Weight initialization in neural nets roughly follows $\mathcal{U}(1/\sqrt{D}, 1/\sqrt{D})$, which will be used to implement random projectors.

Theorem 3 (MLP projectors with random weights enforce pairwise independence.)**.** *Let us consider a MLP composed of alternating random linear layers and elementwise nonlinearities. Then, minimizing the variance and covariance of the output $\boldsymbol{Z}$ enforces pairwise independence between all pairs of variables in the input $\boldsymbol{X}$.*

Proof. Let us consider the last block of such MLP, *i.e.* a fully-connected linear layer followed by an elementwise nonlinearity. According to lemma 1, applying VCReg to the MLP, *i.e.* at the output of the nonlinearity will enforce pairwise independence under corresponding kernel for the input of the nonlinearity, which is also the output of the last linear layer. If the latter is wide enough so that it can be considered orthogonal, Theorem 11 in Comon (1994) ensures that pairwise independence is preserved for the input of the layer. We can then recursively extend the result backward to the whole MLP. If the last MLP layer is a fully-connected linear, then lemma 2 applies and we go back to the preceding elementwise nonlinearity. □

Fig. 5 in Appendix C.1 shows that each layer in the random MLP is recursively VC-regularized from VCReg being applied only at the projector's output. Following Theorem 3, we can expect that wider projectors are beneficial to enforce pairwise independence while adding layers or learning the projector is not necessary; see Section 5 for empirical validation.

Extension to BarlowTwins and W-MSE, and generality of VCReg. Our results focus on VCReg as formulated in VICReg but can in fact be extended to methods that constrain the covariance of $\boldsymbol{Z}$ explicitly, namely BarlowTwins and W-MSE (Balestriero & LeCun, 2022). Indeed, the objective in W-MSE (Equations 3-4 in Ermolov et al. (2021)) is VICReg with explicit constraint on the variance and covariance. Increasing the variance and covariance hyper-parameters in VICReg produces W-MSE hence our results extend seamlessly. The objective from Eq. (7) in BarlowTwins is also similar to VICReg. The derivation, deferred to Appendix B.2, shows that minimizing the constrained form of BarlowTwins objective from Eq. (7) is equivalent to minimizing VICReg's invariance term whilst explicitly constraining the variance covariance terms as in W-MSE; hence our results also hold for BarlowTwins; see Section 5 for empirical validation. In particular we will see that as BarlowTwins explicitly enforces minimum VCReg, it better optimizes HSIC compared to VICReg with standard hyper-parameters. Finally, as opposed to BarlowTwins loss and most SSL methods, VCReg can be used and be beneficial within single branch architectures. We provide such use case in Section 5.

Learned projectors. In state-of-the-art SSL representations, the projector is learned, which is not rigorously covered by Theorem 3. Complementing the study of Bordes et al. (2022), we argue that learning the projector is only crucial to satisfy the invariance criterion since random projectors are sufficient to obtain pairwise independent features, as demonstrated in Section 5. In fact, our experiments show that i) using VICReg, keeping the projector random generally produces lower HSIC of their input i.e. better enforce pairwise independence than when learning the projector's weights, ii) using VICReg, learning the projector to optimize VCReg more strongly than the invariance term reduces performances, and iii) using VCReg, learning the projector e.g. in the later studied ICA setting creates a degenerate representation that does not enforce pairwise independence. We thus conjecture that learning the projector's parameters to mainly minimize VICReg's invariance term leaves the parameters close enough to their random initialization (Jacot et al., 2018) to maintain an accurate estimate of HSIC. In fact, we will see that the wider the projector, the less far away from initialization the parameters have to move, and the better optimized will be HSIC.

4 Related Work

Feature decorrelation. Also known as whitening, feature decorrelation ensures that correlation between each pair of different features in a batch of feature vectors is zero, and that each feature has unit variance. It was originally used as a data pre-processing technique, see e.g. Hyvärinen & Oja (2000), before being extended to deep networks (Cogswell et al., 2015) as a regularizer. In the context of SSL, Hua et al. (2021); Ermolov et al. (2021) find that feature decorrelation helps solving the collapse issue. The former avoids collapse via appropriate Batch Normalization and its decorrelating variant (Huang et al., 2018) in the projector. This variant of Batch Normalization can be seen as the hard constraint counterpart of BarlowTwins. Practically, whitening can be implemented by learning a fully-connected layer Husain & Bober (2019) recovering Principal Component Analysis; as mentioned earlier (Section 3), this differs from VCReg which keeps the layer's parameters random and optimize its input.

Independence criterion for learning features. Enforcing mutual independence to learn a representation has been proposed for example by Schmidhuber (1992). More recently, and in the context of supervised learning, Chen et al. (2019) demonstrated improved training of ResNets by reducing the pairwise dependence in the features at each layer via a combination of Dropout and Batch Normalization. By opposition, VCReg is applied to the projector's output and common SSL projectors do not rely on Dropout; we also show that Batch Normalization is not necessary to reduce pairwise dependence.

HSIC in supervised and self-supervised learning. HSIC-based losses have been employed in supervised learning e.g. see Mooij et al. (2009); Greenfeld & Shalit (2020) to ensure independence between the residual errors and the labels of a task at hand. Kornblith et al. (2019) measure similarity

between two neural representations via HSIC. In SSL, Tsai et al. (2021) showed that a modified version of BarlowTwins loss maximizes HSIC under a linear kernel between the embeddings of two augmented versions of the same sample. In a similar fashion, Li et al. (2021) proposed a SSL framework based on maximizing the dependence between the embeddings of transformations of an image and the image identity via HSIC. Both these works differ from ours, which connects VCReg to the minimization of HSIC between pairs of features in the representation.

5 EXPERIMENTS

We first put in evidence pairwise independence properties emerging in learned visual representations, before validating and exploiting our theoretical findings. Finally, we ablate the SSL context and demonstrate that the composition of a SSL-like random projector with VCReg induces enough independence to perform ICA. Importantly, none of the experiments with VCReg use Dropout, and Batch Normalization is not used when the projector is random. Hence, pairwise independence cannot be attributed to those two techniques as opposed to Chen et al. (2019); Hua et al. (2021).

5.1 PAIRWISE INDEPENDENCE EMERGES IN MOST VISUAL REPRESENTATIONS

Setup. In these experiments, we track two metrics for statistical independence of the components in the learned representation during ResNet50 training on ImageNet with popular SSL frameworks, as well as a supervised baseline. The first metric, HSIC (Gretton et al., 2005a), tracks pairwise independence. The second metric, dHSIC (Pfister et al., 2018), tracks mutual independence. As explained in Section 2.1, neither HSIC nor dHSIC scale to the full representation (2048 components). Therefore, we instead rely on proxys: for pairwise independence, we compute and average HSIC on all pairs of the first n components of the representation. For mutual independence, we sample sets of n components for which we display dHSIC. We set $n = 10$ and compute these statistics on the whole ImageNet validation set. We expect both metrics to decrease during the training, which would suggest decreasing dependence among the representations. Although statistical tests are available for both metrics, we track bare HSIC and dHSIC as they are continuous. However, we perform HSIC tests in this first serie of experiments so that the bare values can be linked to concrete portions of independent pairs. The width of the Gaussian kernel in HSIC is scaled for each representation (see 2.1) thus allowing for comparison between methods. See Appendix E for the detailed setup.

Results. Figure 1 shows that different learning frameworks implicitly optimize pairwise independence of the variables in the representation throughout the training (left), and at some point mutual independence (right). As expected from Section 3, HSIC is continuously optimized in Barlow Twins and VICReg (it increases for DINO)[1], and Barlow Twins better enforces pairwise independence since the covariance matrix is constrained. Precise values along with HSIC independence tests at level $\alpha = 0.5$ and corresponding test accuracies can be found in Table 1. Lower HSIC values do not necessarily entail more successful tests: it is possible to allocate low HSIC to sufficiently many pairs while having an overall larger HSIC. For example, Supervised has less independent pairs in spite of an overall smaller HSIC. Table 1 also suggests that pairwise independence is not predictive of test accuracy. Although it seems to be a desirable property, it is not sufficient to yield a good representation: for example, DINO achieves better test accuracy than BarlowTwins while having higher HSIC. We study this trade-off in the next subsection. Although we observe decreasing dHSIC, *i.e.* improvement of mutual independence in the representations, all methods quickly reach a floor and do not improve further although it is lower for Barlow Twins and VICReg. Hence, we cannot conclude that this property is properly optimized and do not study it in subsection 5.2, but conduct experiments to better evaluate mutual independence obtained by VCReg in subsection 5.3.

5.2 PROJECTOR CHARACTERISTICS FOSTERING OR HURTING PAIRWISE INDEPENDENCE

Setup. These experiments validate the results from Section 3 by studying which projector characteristics helps or hurt HSIC optimization when learning visual representations. The setup is the same as above except that dHSIC is not used anymore and the projector may be replaced by projectors

[1]HSIC is continuously optimized for SimCLR as well: this could be explained in light of Garrido et al. (2022), which connect VICReg and SimCLR, and is omitted here for conciseness.

[2]As opposed to other methods here, DINO relies on multicrop, which partly explains the performance gap.

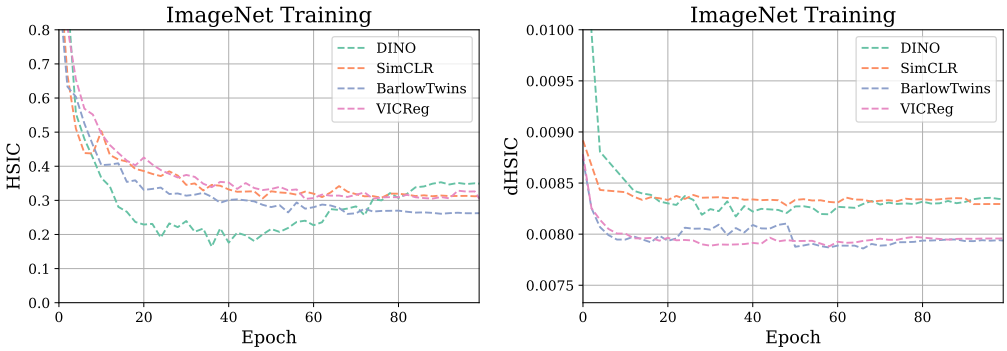

Figure 1: Overall, HSIC and dHSIC of the encoder output improves for all methods during training. VCReg based methods have smoother and/or further optimization of HSIC and dHSIC.

Table 1: Measure of pairwise dependence (HSIC), pairwise independence testing, and test accuracy for popular SSL or supervised representations averaged on multiple subsets of features.

Method (100 ep.)	HSIC ↓	% of indep. pairs ↑	ImageNet Top1 ↑
BarlowTwins (Zbontar et al., 2021)	0.30 ± 0.03	38 ± 2	67.8
VICReg (Bardes et al., 2022)	0.39 ± 0.03	34 ± 2	68.4
SimCLR (Chen et al., 2020)	0.40 ± 0.03	36 ± 1	67.7
DINO (Caron et al., 2021)	0.38 ± 0.02	35 ± 3	70.4[2]
Supervised (Wightman et al., 2021)	0.20 ± 0.05	33 ± 3	78.1

introduced in Section 3: an elementwise nonlinear projector from lemma 1, random linear projectors of various width from lemma 2, and a MLP taking the representation ($D = 2048$) as input, and outputting an embedding of size P, with two hidden layers of width P and ReLUs. The MLP can be random as in Theorem 3 or learned as in Section 5.1. We apply an invariance loss along with VCReg on the output of the projectors and scale the covariance coefficient with the size of the projector for fair comparison. See Appendix E for the detailed setup.

Results. Figure 2 (top left) demonstrates that nonlinear elementwise projectors, random linear projectors and random MLPs with one layer achieve lower HSIC than learned MLPs, in particular when resampled so as to get better HSIC estimates. Hence, learning the projector is not necessary to obtain pairwise independence. As expected in Section 3, for both theoretical and classical projectors, increasing width yields lower HSIC while increasing depth hurts HSIC, see Figure 2. Table 4 in Appendix C.1 shows that increasing the weight of VCReg does not improve HSIC: a possible explanation is that the projector would otherwise start to optimize too much for VCReg. This intuition will be backed by Experiments 5.3. **Modifying Batch Normalization to get closer to a universal kernel in lemma 1 yields significant test accuracy gains for VICReg over Bardes et al. (2022)** (Figure 2, top right); the Batch Normalization employed (see Figure 7) enables explicit control of the statistics standard deviation and prevents gradient to flow through them. Finally, HSIC is not correlated to test accuracy across all methods but the correlation is true within the same method, see Figure 2 (top right).

Discussion. As expected from Section 3, Batch Normalization in the projector is not required to enforce pairwise independence: Figure 2 (top right) shows that removing BN improves HSIC while maintaining test accuracy, which could be explained by the fact that each activation in the projector is implicitly regularized by VCReg (Figure 5), alleviating the need for BN. This could also explain why adding layers in detrimental to HSIC: implicit VCReg of the activations, an assumption of Theorem 3, will be more and more loosely enforced. Although projectors from Section 3 learn representations with better pairwise independence properties than classical SSL methods, most of them are not as competitive in terms of test accuracy (Fig. 2, top right). This complements the view of Appalaraju et al. (2020); Bordes et al. (2022), which argue that the projector requires some learning capacity to filter out information that is irrelevant to the invariance criterion. Our experiments suggest that projector capacity should rather be increased via width than via depth: adding layers can be

detrimental to HSIC as seen above but also to test accuracy. Appalaraju et al. (2020); Zbontar et al. (2021) also observe that the performance saturates for projectors with more than three layers.

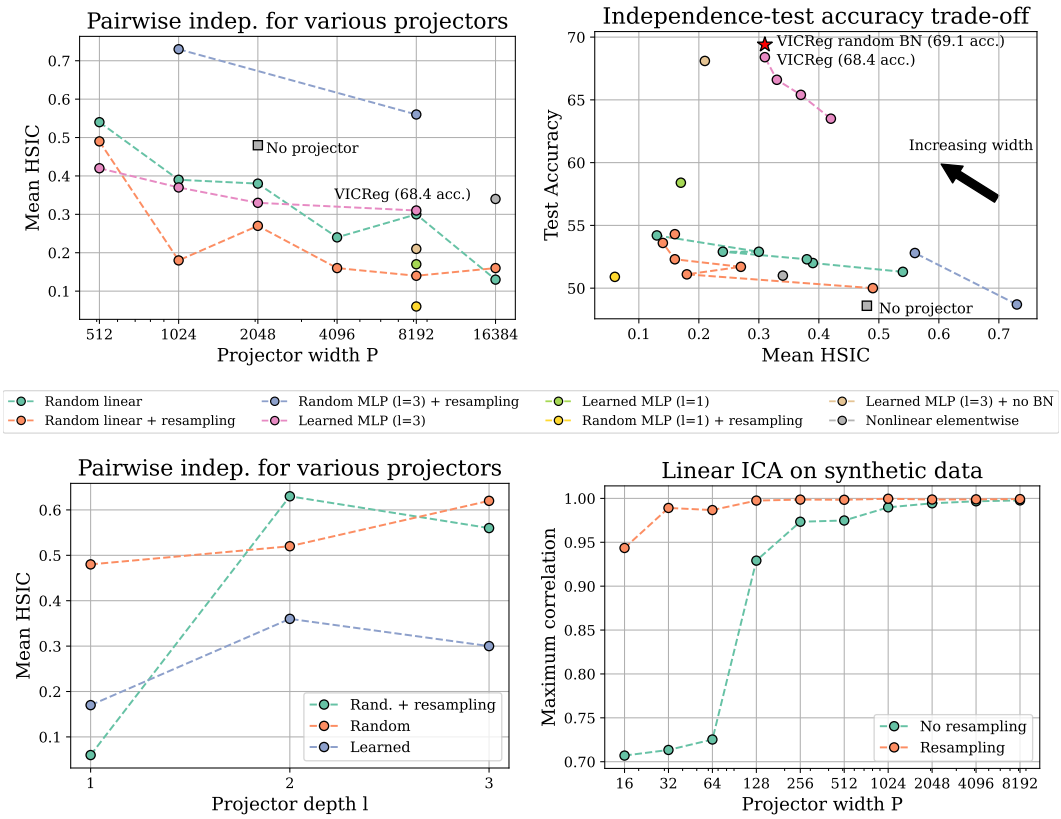

Figure 2: Top left: wider projectors and resampling both yield more pairwise independence for the encoder output. Bottom left: increasing depth hurts HSIC for random or learned projectors. Top right: trade-off between independence of the representation and test accuracy (width $D = 8192$). Bottom right: ICA experiments. Resampling and increased width improve the quality of the reconstruction.

5.3 Ablation: Independent Component Analysis (ICA) with VCReg

The following experiments aim to demonstrate that our findings hold outside of SSL. To this end, we show that VCReg of a SSL-like projector's output induces enough pairwise independence to solve ICA. The goal of ICA (Comon, 1994) is to find a transformation of a random vector $\boldsymbol{Y}$[3] which minimizes the statistical dependence of its components. Its simplest instance is linear ICA, and is motivated by problems such as the cocktail-party, where $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ typically results from a linear transformation of independent sources $\boldsymbol{S}$ (*e.g* overlapping voices) one wants to recover. Formally, $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{A}$, with $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ an unknown mixing matrix that can therefore not be inverted. Instead, the linear ICA approach searches for $\boldsymbol{M} \in \mathbb{R}^{D \times D}$ such that $\boldsymbol{M}\boldsymbol{Y} = \boldsymbol{S}$ by maximizing the statistical *mutual* independence between the components of $\boldsymbol{M}\boldsymbol{Y}$. Being able to recover $\boldsymbol{S}$ is a property known as identifiability. In this setting, enforcing *pairwise* independence in the components of $\boldsymbol{M}\boldsymbol{Y}$ is generally sufficient to recover the true signal s (Comon (1994), Theorem 11).

VC regularized projectors solve linear ICA. Since pairwise independence is sufficient to solve linear ICA, VCReg of a random projector's output should be able to recover $\boldsymbol{S}$. Our model can be seen on Figure 3: whitened batches of mixtures $\boldsymbol{Y}$ are fed to an encoder f. Here, f is a linear transformation $\boldsymbol{M}$ described above. The output $\boldsymbol{X} = f(\boldsymbol{Y})$ is then fed to a projector g, and VCReg is applied to the output $\boldsymbol{Z} = g(\boldsymbol{X})$ covariance matrix. As in Experiments 5.2, the projector is randomly resampled at each gradient step to get better HSIC estimates. This setting corresponds to one branch of a VICReg network i) with a linear projection M as encoder instead of a neural network, ii) with a

[3]For simplicity, $\boldsymbol{Y}$ directly denotes N empirical D-dimensional observations.

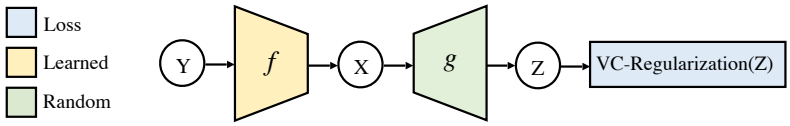

Figure 3: Linear ICA model expressed from VCReg of the projector's output.

Table 2: Maximum correlation (between true and reconstructed sources (the higher the better ↑). The projector performs competitive reconstruction in the linear setting (left), where pairwise independence is sufficient, but not in PNL (right) where it is not.

Linear mixture			Post Non Linear mixture		
Method	Synthetic data	Audio data	Method	Synthetic data	Audio data
Whitening	0.8074	0.9876	Whitening	0.7981	0.9046
FastICA	0.9998	1.0	FastICA	0.8311	0.8989
Anica	0.9987±6.5e-4	0.9996±4.9e-4	Anica	0.9794±53e-4	0.9929±18e-4
VCReg	0.9986±8.2e-4	0.9936±64e-4	VCReg	0.8465±142e-4	0.8706±376e-4

random projector, and iii) without invariance criterion. We perform ICA on two datasets, a synthetic one (Brakel & Bengio, 2017) with 6 sources among which 2 are noise, and a real audio dataset (Kabal, 2002) with 3 sources among which one is noise also used in Brakel & Bengio (2017). The evaluation metric is the maximum correlation between the true and reconstructed sources. As this metric is not available without ground truth, we select the model with lowest dHSIC for $\boldsymbol{X}$, which can be perfectly evaluated here since the number of sources is small. Our model recovers the sources, and is even competitive with methods specifically designed for linear ICA such as Fast ICA (Hyvärinen & Oja, 2000) or Anica (Brakel & Bengio, 2017), see Table 2. Both increasing the width of the projector and resampling it at each step (especially for smaller projectors) improve the reconstruction as can be seen in Figure 2 (bottom right). This is in line with our findings in Section 3 and Experiment 5.2.

VC regularized projectors do not solve nonlinear ICA. Experiment 5.1 suggests that mutual independence also improves during training, although not as clearly as pairwise independence. Hence, one could ask whether VCReg also enforces the former enough to solve nonlinear ICA. To test this hypothesis, we apply VCReg to a particular case of nonlinear ICA which allows identifiability but does not have equivalence between pairwise and mutual independence: the post-nonlinear mixture (PNL) (Taleb & Jutten, 1999). In PNL, the sources are linearly mixed before being fed to elementwise nonlinear functions. For these experiments, and following Brakel & Bengio (2017), our encoder is a MLP. During our first experiments, we observed an informational collapse of the encoder, which produces seemingly mutually independent variables with very poor reconstruction of the sources. To alleviate this issue, we add a decoder taking $\boldsymbol{X}$ as input and reconstructing $\boldsymbol{Y}$. Figure 8 in Appendix E shows our modified setup. We compare VCReg to FastICA and Anica. Although FastICA is not meant to solve the nonlinear case, it remains an interesting baseline. Table 2 shows that our model fails to recover the sources, as it does only slightly better than FastICA in the synthetic case. Hence, although mutual independence increases during training, VCReg does not optimize it enough to solve nonlinear ICA. We propose a simple explanation for this limitation. Indeed, each feature in $\boldsymbol{Z}$ is a nonlinear function of all features in $\boldsymbol{X}$. Hence, it is not possible to completely decorrelate two components of $\boldsymbol{Z}$ as they both contain the same set of features. It is still possible to improve mutual independence to some extent since, in practice, only parts of the inputs are considered at once by nonlinear mappings such as neural networks (Erhan et al., 2009; Adebayo et al., 2018).

Learning the projector does not enforce pairwise independence. ICA experiments done with a learned projector fail to recover independent sources: this further demonstrates that learning the projector to specifically optimize the VCReg is counter-productive, and that in the context of SSL, learning the projector is rather useful for satisfying the invariance criterion, which is absent in the ICA experiments. VCReg is rather optimized by the encoder f.

6 Conclusion

Although pairwise independence in the representation emerges from many SSL criteria, it can be precisely explained for frameworks using VCReg. This study also shows that randomizing the nonlinearities yields significant gains on very competitive methods. Wider and shallower projectors should also be explored when increasing the size of the projector, a task we leave for future work since this raises computational issues. Our experiments on ICA not only confirm our findings but expand the scope of VCReg outside of SSL. Moreover, VCReg seems to enforce pairwise independence only, and future work could consist in ensuring better optimization of pairwise and/or mutual independence.

7 REPRODUCIBILITY STATEMENT

For theoretical results, explanations on assumptions are provided in Section 3 and proofs are displayed either in the main paper or in Appendix B.1. The results are verified experimentally in Section 5.

Implementations of the algorithms used (HSIC, dHSIC, Randomized Batch Normalization, Linear ICA, and PNL ICA) are provided in Appendix D, and our code will be released if the paper is accepted.

Our experimental setup is detailed either in the main paper or in Appendix E. The latter include architectural details, augmentations, optimizer with learning rate and batch size, and grid search for hyper-parameters. The numbers used to produce the Figures in 5 are provided in C.1.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Srikar Appalaraju, Yi Zhu, Yusheng Xie, and István Fehérvári. Towards good practices in self-supervised representation learning. *arXiv preprint arXiv:2012.00868*, 2020.

Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research (JMLR)*, 2002.

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.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *International Conference on Learning Representations (ICLR)*, 2022.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Improving deep networks generalization by removing their head. *arXiv preprint arXiv:2206.13378*, 2022.

Philemon Brakel and Yoshua Bengio. Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*, 2017.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Di Chen and Jeff M Phillips. Relative error embeddings of the gaussian kernel distance. In *International Conference on Algorithmic Learning Theory*, pp. 560–576. PMLR, 2017.

Guangyong Chen, Pengfei Chen, Yujun Shi, Chang-Yu Hsieh, Benben Liao, and Shengyu Zhang. Rethinking the usage of batch normalization and dropout in the training of deep neural networks. *arXiv preprint arXiv:1905.05928*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

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

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Romain Cosentino, Anirvan Sengupta, Salman Avestimehr, Mahdi Soltanolkotabi, Antonio Ortega, Ted Willke, and Mariano Tepper. Toward a geometrical understanding of self-supervised contrastive learning. *arXiv preprint arXiv:2205.06926*, 2022.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning (ICML)*, 2021.

Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann Lecun. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022.

Bernhard Goebel, Zaher Dawy, Joachim Hagenauer, and Jakob C Mueller. An approximation to the distribution of finite sample size mutual information estimates. In *IEEE International Conference on Communications, 2005. ICC 2005. 2005*, volume 2, pp. 1102–1106. IEEE, 2005.

Daniel Greenfeld and Uri Shalit. Robust learning with the hilbert-schmidt independence criterion. In *International Conference on Machine Learning (ICML)*, 2020.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005a.

Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, Bernhard Schölkopf, et al. Kernel methods for measuring independence. *Journal of Machine Learning Research (JMLR)*, 2005b.

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 (NeurIPS)*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Syed Sameed Husain and Miroslaw Bober. Remap: Multi-layer entropy-guided pooling of dense cnn features for image retrieval. *IEEE Transactions on Image Processing*, 28(10):5201–5213, 2019.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Peter Kabal. Tsp speech database, 2002. McGill University, Database Version, 1(0):09-02.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Sue E Leurgans, Rana A Moyeed, and Bernard W Silverman. Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(3): 725–740, 1993.

Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zejian Li, Yongchuan Tang, Wei Li, and Yongxing He. Learning disentangled representation with pairwise independence. In *Proceedings of the AAAI conference on artificial intelligence*, 2019.

Edward L Melnick and Aaron Tenenbein. Misspecifications of the normal distribution. *The American Statistician*, 36(4):372–373, 1982.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research (JMLR)*, 2006.

Joris Mooij, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf. Regression by dependence minimization and its application to causal inference in additive noise models. In *International Conference on Machine Learning (ICML)*, 2009.

Edith Mourier. Eléments aléatoires dans un espace de banach. In *Annales de l'institut Henri Poincaré*, volume 13, pp. 161–244, 1953.

Niklas Pfister, Peter Bühlmann, Bernhard Schölkopf, and Jonas Peters. Kernel-based tests for joint independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1): 5–31, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

Alfréd Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451, 1959.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4 (6):863–879, 1992.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pp. 13–31. Springer, 2007.

Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research (JMLR)*, 2001.

Yitong Sun, Anna Gilbert, and Ambuj Tewari. On the approximation properties of random relu features. *arXiv preprint arXiv:1810.04374*, 2018.

Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on signal Processing*, 47(10):2807–2820, 1999.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning (ICML)*, 2021.

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

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

A FURTHER BACKGROUND ON VCREG METHODS

BarlowTwins. Zbontar et al. (2021) propose a slightly different approach based on regularizing $\boldsymbol{C}$, the $P \times P$ cross-correlation matrix between $\boldsymbol{Z}_{\text{left}}$ and $\boldsymbol{Z}_{\text{right}}$ by optimizing

$$\mathcal{L}_{\text{BT}} = \sum_{k=1}^{K}((\boldsymbol{C})_{k,k} - 1)^2 + \alpha \sum_{k' \neq k} (\boldsymbol{C})_{k,k'}^2 \ . \tag{7}$$

Here, the leftmost term corresponds to regularizing the cross-correlation of the same feature in the two views to be unit, while the rightmost term regularizes the cross-correlation between pairs of different features in the two views. Importantly, $(\boldsymbol{C})_{i,j}$ falls back to measuring the cosine similarity between the ith column of $\boldsymbol{Z}_{\text{left}}$ and the jth column of $\boldsymbol{Z}_{\text{right}}$ i.e. $(\boldsymbol{C})_{i,j} = \frac{\langle (\boldsymbol{Z}_{\text{left}})_{.,i}, (\boldsymbol{Z}_{\text{right}})_{.,j} \rangle}{\|(\boldsymbol{Z}_{\text{left}})_{.,i}\|_2 \|(\boldsymbol{Z}_{\text{right}})_{.,j}\|_2}$. Hence, the leftmost term is also an invariance term: all features must be similar for both views.

W-MSE. Ermolov et al. (2021) use the following loss

$$\begin{aligned} &\min\ 2 - 2\frac{\langle \boldsymbol{Z}_{\text{left}}, \boldsymbol{Z}_{\text{right}} \rangle}{\|\boldsymbol{Z}_{\text{left}}\|_2 \|\boldsymbol{Z}_{\text{right}}\|_2} \\ &\text{s.t. } \text{Cov}(\boldsymbol{Z}_{\text{left}}) = Id, \text{Cov}(\boldsymbol{Z}_{\text{right}}) = Id. \end{aligned}$$

where the cosine similarity can be replaced by the Euclidean distance.

B PROOFS

B.1 PROOF OF LEMMA 2: RANDOM LINEAR PROJECTORS MAXIMIZE PAIRWISE INDEPENDENCE

Proof. Let's consider the linear regime for which $g((\boldsymbol{X})_{:,i}) = (\boldsymbol{X})_{:,i}\boldsymbol{w}_i^T$ with orthogonal weights i.e. $\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle = 1_{\{i=j\}}$. This assumption is realistic since we assume that $\boldsymbol{w} \in \mathbb{R}^K$ with K very large, and that those are randomly sampled. From that, we then have

$$\begin{aligned} \sum_{i \neq j} \text{HSIC}((\boldsymbol{X})_{:,i}, (\boldsymbol{X})_{:,j}) =& \frac{1}{(N-1)^2} \sum_{i \neq j} \text{Tr}(g((\boldsymbol{X})_{:,i}) g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j}) g((\boldsymbol{X})_{:,j})^T \boldsymbol{H}) \\ =& \frac{1}{(N-1)^2} \sum_{i \neq j} \|g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j})\|_F^2, \end{aligned}$$

we will now push the sum inside the norm by considering the following equality:

$$\|\sum_{i \neq j} f(i,j)\|_F^2 = \sum_{i \neq j} \|f(i,j)\|_F^2 + \sum_{i \neq j} \sum_{k \neq \ell, (i,j) \neq (k,\ell)} \text{Tr}\left(f(i,j)^T f(k,\ell)\right) = \sum_{i \neq j} \|f(i,j)\|_F^2,$$

since in our case

$$\begin{aligned} f(i,j) &= g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j}) = \boldsymbol{w}_i (\boldsymbol{X})_{:,i}^T \boldsymbol{H} (\boldsymbol{X})_{:,j} \boldsymbol{w}_j^T, \\ f(i,j)^T f(k,\ell) &= \boldsymbol{w}_j (\boldsymbol{X})_{:,j}^T \boldsymbol{H} (\boldsymbol{X})_{:,i} \boldsymbol{w}_i^T \boldsymbol{w}_k (\boldsymbol{X})_{:,k}^T \boldsymbol{H} (\boldsymbol{X})_{:,\ell} \boldsymbol{w}_\ell^T, \\ \text{Tr}(f(i,j)^T f(k,\ell)) &= 1_{\{i=k \wedge j=\ell\}} (\boldsymbol{X})_{:,j}^T \boldsymbol{H} (\boldsymbol{X})_{:,i} (\boldsymbol{X})_{:,k}^T \boldsymbol{H} (\boldsymbol{X})_{:,\ell}, \end{aligned}$$

leading to

$$\begin{aligned} \sum_{i \neq j} \text{HSIC}((\boldsymbol{X})_{:,i}, (\boldsymbol{X})_{:,j}) =& \frac{1}{(N-1)^2} \|\sum_{i \neq j} g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,j})\|_F^2 \\ =& \frac{1}{(N-1)^2} \|(\sum_i g((\boldsymbol{X})_{:,i}))^T \boldsymbol{H} (\sum_j g((\boldsymbol{X})_{:,j})) - \sum_i g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,i})\|_F^2 \\ =& \|\frac{1}{N-1} (\sum_i g((\boldsymbol{X})_{:,i}))^T \boldsymbol{H} (\sum_j g((\boldsymbol{X})_{:,j})) - \text{diag}(\text{Cov}(\boldsymbol{X}\boldsymbol{W}))\|_F^2 \\ =& \|\text{Cov}(\boldsymbol{X}\boldsymbol{W}) - \text{diag}(\text{Cov}(\boldsymbol{X}\boldsymbol{W}))\|_F^2 \\ =& \sum_{i \neq j} \text{Cov}(\boldsymbol{X}\boldsymbol{W})_{i,j}^2, \end{aligned}$$

since we assume that all columns of $\boldsymbol{X}$ have same variance and that $\boldsymbol{W}$ is orthogonal we have for the pre-last equality

$$\frac{1}{N-1}\sum_i g((\boldsymbol{X})_{:,i})^T \boldsymbol{H} g((\boldsymbol{X})_{:,i}) = \mathrm{Var}((\boldsymbol{X})_{:,1})\sum_i \boldsymbol{w}_i \boldsymbol{w}_i^T = \mathrm{Var}((\boldsymbol{X})_{:,1}),$$

and for the last equality we use the fact that since g is linear, $\sum_i g((\boldsymbol{X})_{:,i}) = \sum_i (\boldsymbol{X})_{:,i}\boldsymbol{w}_i^T = \boldsymbol{X}\boldsymbol{W}$ with $\boldsymbol{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_K]^T$.

□

B.2 On the equivalence between VICReg and BarlowTwins objectives

We can express Barlow Twins objective as:

$$\begin{aligned}&\min \sum_{k=1}^{K}(\mathrm{Cov}(\boldsymbol{Z}_{\text{left}}, \boldsymbol{Z}_{\text{right}})_{k,k} - 1)^2 + \alpha \sum_{k' \neq k} \mathrm{Cov}(\boldsymbol{Z}_{\text{left}}, \boldsymbol{Z}_{\text{right}})^2_{k,k'} \\ &\text{s.t. } \mathrm{Cov}(\boldsymbol{Z}_{\text{left}}) = \boldsymbol{I}, \mathrm{Cov}(\boldsymbol{Z}_{\text{right}}) = \boldsymbol{I}.\end{aligned}$$

Assuming $\boldsymbol{Z}_{\text{left}}^T \boldsymbol{Z}_{\text{left}} = \boldsymbol{I}, \boldsymbol{Z}_{\text{right}}^T \boldsymbol{Z}_{\text{right}} = \boldsymbol{I}$ i.e. perfect minimization of the variance and covariance terms, we have

$$C_{i,j} = \frac{\langle (\boldsymbol{Z}_{\text{left}})_{:,i}, (\boldsymbol{Z}_{\text{left}})_{:,j}\rangle}{\|(\boldsymbol{Z}_{\text{left}})_{:,i}\|_2 \|(\boldsymbol{Z}_{\text{left}})_{:,j}\|_2} = \frac{1}{2}\langle (\boldsymbol{Z}_{\text{left}})_{:,i}, (\boldsymbol{Z}_{\text{right}})_{:,j}\rangle = -\|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2 - 1,$$

and thus

$$\sum_i (C_{i,i} - 1)^2 = \sum_i \|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{right}})_{:,i}\|_2^4 = \|\boldsymbol{Z}_{\text{left}} - \boldsymbol{Z}_{\text{right}}\|_F^4 \propto I(\boldsymbol{Z}_{\text{left}}, \boldsymbol{Z}_{\text{right}}),$$

so we see that we recover the invariance loss exactly with the diagonal terms of BarlowTwins. Now the only thing left to show is that this is actually enough to minimize to also minimize the off-diagonal terms. To see that, let's follow the derivations below

$$\begin{aligned}\sum_{i\neq j} C_{i,j}^2 =& \sum_{i\neq j}(1 - \frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2)^2 \\ =& \sum_{i\neq j}(1 - \frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j} + (\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2)^2 \\ =& \sum_{i\neq j}\Big(1 - \frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j}\|_2^2 - \frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2 \\ &- \langle (\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j}, (\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\rangle\Big)^2 \\ =& \sum_{i\neq j}\Big(-\frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2 - \langle (\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j}, (\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\rangle\Big)^2 \\ =& \sum_{i\neq j}\Big(\frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2 + \langle (\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j}, (\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\rangle\Big)^2 \\ \leq& \sum_{i\neq j}\Big(\frac{1}{2}\|(\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2 + \|(\boldsymbol{Z}_{\text{left}})_{:,i} - (\boldsymbol{Z}_{\text{left}})_{:,j}\|_2^2 \|(\boldsymbol{Z}_{\text{left}})_{:,j} - (\boldsymbol{Z}_{\text{right}})_{:,j}\|_2^2\Big)^2 \\ =& \frac{25(K-1)}{4}\|\boldsymbol{Z}_{\text{left}} - \boldsymbol{Z}_{\text{right}}\|_2^4,\end{aligned}$$

hence minimizing the BarlowTwins loss with the explicit whitening constraint is equivalent to minimizing VICReg with explicit whitening constraint.

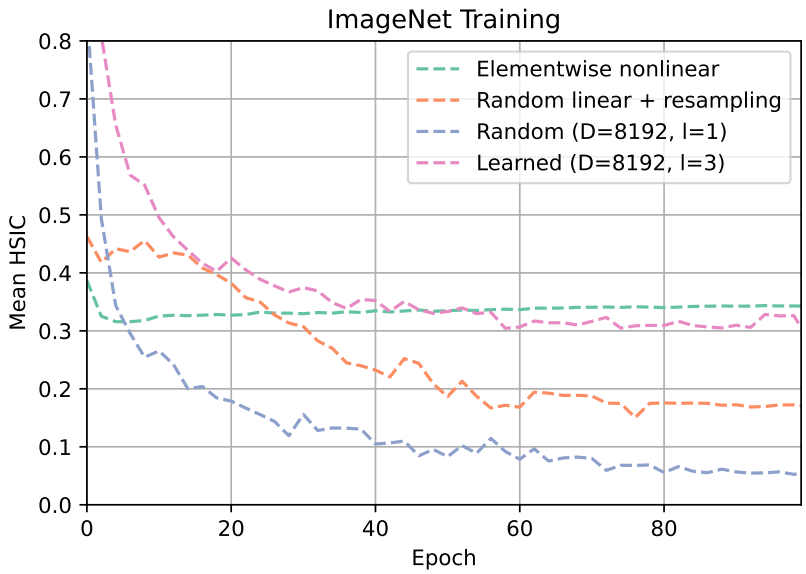

Figure 4: The projectors proposed in Section 3 better optimize HSIC than their classical counterpart.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 HSIC OF REPRESENTATIONS LEARNED FROM IMAGENET

Table 3: Measure of pairwise dependence (the lower the better) and test accuracy for the projectors studied in Section 3. L is the number of nonlinear projections in Lemma 1, D is the width of the projector and l is the number of layers.

	Resampling		No resampling	
Projector type	Mean HSIC ↓	ImageNet Top1 ↑	Mean HSIC ↓	ImageNet Top1 ↑
No projector	N.A.	N.A.	0.48	48.6
Elementwise nonlinear ($L = 8$)	N.A.	N.A.	0.34	51.0
Random linear ($D = 16384$)	0.16	54.3	0.13	54.2
Random linear ($D = 8192$)	0.14	53.6	0.30	52.9
Random linear ($D = 4096$)	0.16	52.3	0.24	52.9
Random linear ($D = 2048$)	0.18	51.7	0.38	52.3
Random linear ($D = 1024$)	0.27	51.1	0.39	52.0
Random linear ($D = 512$)	0.49	50.0	0.54	51.3
Random ($D = 8192$, $l = 1$)	0.06	50.9	0.48	53.6
Random ($D = 8192$, $l = 2$)	0.63	53.3	0.52	53.4
Random ($D = 8192$, $l = 3$)	0.56	52.8	0.62	54.9
Learned ($D = 8192$, $l = 1$)	0.17	58.4	N.A.	N.A.
Learned ($D = 8192$, $l = 2$)	0.36	65.1	N.A.	N.A.
Learned ($D = 8192$, $l = 3$)	0.30	68.4	N.A.	N.A.
Same without BN	0.22	68.1	N.A.	N.A.
Same with Random BN	0.31	69.1	N.A.	N.A.

Table 4: VICReg with 2048-2048-2048 projector and varying covariance coefficient. Increasing this coefficient does not necessarily decreases HSIC. Indeed, the projector may start to optimize too much for the VC criterion thus getting trapped in bad solutions.

Covariance coeff	Mean HSIC ↓	ImageNet Top1 ↑
2	0.33	66.6
4	0.42	66.7
6	0.37	66.6
8	0.29	66.3
10	0.28	66.5
12	0.37	66.2

Table 5: VICReg with learned MLP projector and varying sizes. The wider the projector, the lower HSIC, which conforms to our theoretical results in Section 3.

Covariance coeff	Mean HSIC ↓	ImageNet Top1 ↑
512	0.42	63.5
1024	0.37	65.4
2048	0.33	66.6
8192	0.31	68.4

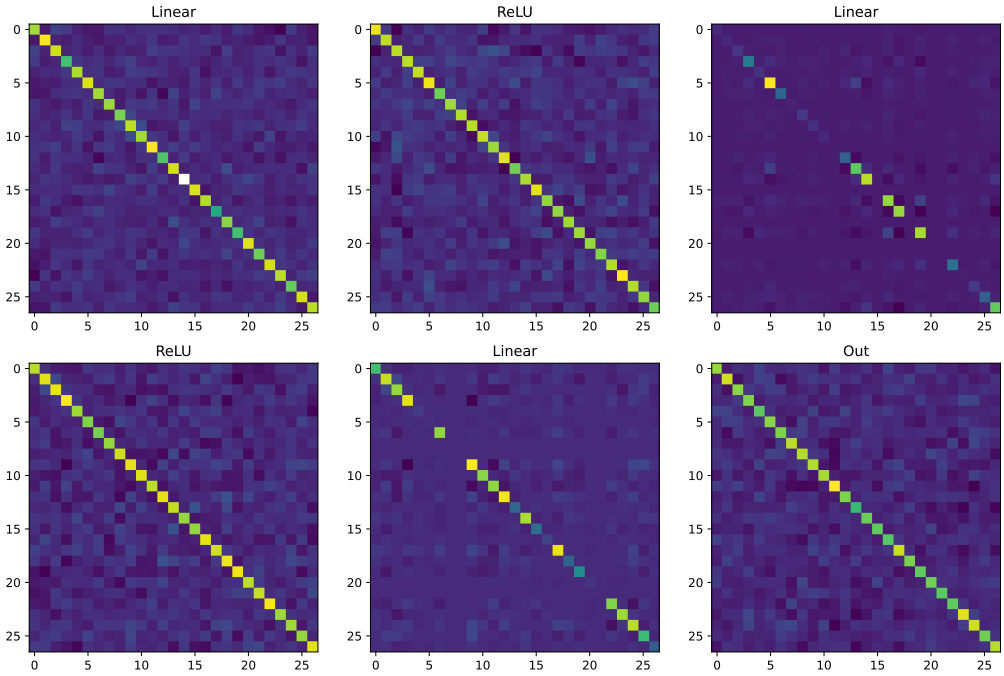

Figure 5: The covariance matrices for some features of each activation in a random VICReg projector (same as in Bardes et al. (2022) with width 8192) before corresponding hidden layer are close to diagonal, suggesting that each hidden layer output is implicitly VC-regularized.

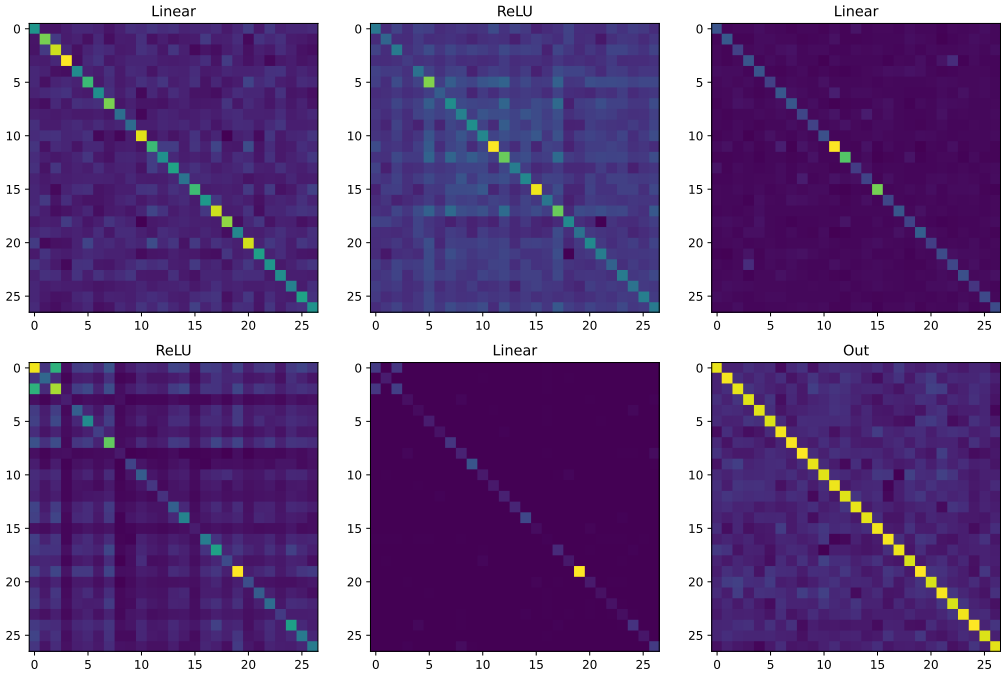

Figure 6: The covariance matrices for some features of each activation in a learned VICReg projector (same as in Bardes et al. (2022) with width 8192) before corresponding hidden layer are close to diagonal, suggesting that each hidden layer output is implicitly VC-regularized.

D IMPLEMENTATIONS

D.1 HSIC EQ. (2)

```
def GaussianKernelMatrix(X, sigma):
    pairwise_distances = torch.cdist(X, X)
    return torch.exp( -pairwise_distances / (2 * sigma**2))

def HSIC(X_1, X_2, sigma_1, sigma_2):
    N = x.size(0) # batch size, should be the same for X_1 and X_2
    K_1 = GaussianKernelMatrix(X_1, sigma_1) # Gaussian kernel matrix of
    X_1 with bandwidth sigma_1
    K_2 = GaussianKernelMatrix(X_2, sigma_2) # Gaussian kernel matrix of
    X_2 with bandwidth sigma_2
    H = torch.eye(N) - 1.0 / N  # centering matrix
    HSIC = torch.trace(K_1 @ H @ K_2 @ H)/((N-1)**2)
    return HSIC
```

In our study, the bandwidth σ of the Gaussian kernel is determined by the median of the distribution of pairwise euclidean distances between samples.

D.2 DHSIC EQ. (4)

dHSIC can be estimated given empirical samples $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_d$ with respective kernel matrices $\boldsymbol{K}_1, \ldots, \boldsymbol{K}_d$ as:

$$\mathrm{dHSIC}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_d) = \frac{1}{N^2} \sum_{i,j} (\bigodot_{k=1}^{d} \boldsymbol{K}_k)_{i,j} + \frac{1}{N^{2d}} \prod_{k=1}^{d} \sum_{i,j} (\boldsymbol{K}_k)_{i,j} - \frac{2}{N^{d+1}} \sum_{i} \bigodot_{k=1}^{d} \sum_{j} (\boldsymbol{K}_k)_{i,j}.$$

When $d = 2$, the first term corresponds to a biased HSIC estimator. The implementation of dHSIC is given by:

```
import GaussianKernelMatrix # defined above

def dHSIC(X, sigma):
    length = X.shape(0)
    term_1 = 1.0
    term_2 = 1.0
    term_3 = 2.0 / length
    for j in range(D):
        K_j = GaussianKernelMatrix(X[j])
        term_1 = torch.mul(term_1, K_j)
        term_2 = 1.0 / length / length / term_2 * torch.sum(K_j)
        term_3 = 1.0 / length * term_3 * K_j.sum(axis=0)

    term_1 = (1.0 / length) ** 2 * torch.sum(term_1)
    term_3 = torch.sum(term_3)

    return term_1 + term_2 - term_3 # the three terms of the estimator
```

As these evaluations already take a few minutes, scaling it to significantly larger portions of components would be impractical.

D.3 RANDOMIZED BATCH NORMALIZATION

```
class RandBatchNorm1d(torch.nn.BatchNorm1d):
    def forward(input):
        # the mean and variance term in classical batch norm are
        # randomized
        m = torch.randn_like(m) * self.estimator_mean.std +
            self.estimator_mean.mean
        std = torch.distributions.Gamma(
            self.estimator_std.mean.square() / self.estimator_std.var,
            self.estimator_std.mean / self.estimator_std.var,
            ).sample()
```

```
        return self.weight * (input - m) / torch.sqrt(std.square_() +
    self.eps) + self.bias
```

D.4 LINEAR ICA MODEL 3

```
for p in projector.parameters(): # freeze the projector
        p.requires_grad = False

for y in loader:
    x = encoder(y) # matrix multiplication with M

    z = projector(x) # compute embedding

    C = torch.cov(z) # covariance matrix
    loss = torch.MSE(C, torch.identity()) # VCReg loss
    loss.backward()
    optimizer.step()

    projector.__init__()  # to resample the projector
    for p in projector.parameters(): # freeze the projector
        p.requires_grad = False
```

D.5 NONLINEAR ICA MODEL 8

The implementation of our nonlinear ICA model 8 differs from 3 by the addition of a reconstruction constraint:

```
for p in projector.parameters(): # freeze the projector
        p.requires_grad = False

for y in loader:
    x = encoder(y) # MLP encoder

    z = projector(x) # compute embedding

    y_rec = reconstructor(x) # MLP reconstructor

    C = torch.cov(z) # covariance matrix
    loss = torch.MSE(C, torch.identity()) + lmda * torch.MSE(y, y_rec) #
    VCReg loss with reconstruction
    loss.backward()
    optimizer.step()

    projector.__init__()  # to resample the projector
    for p in projector.parameters(): # freeze the projector
        p.requires_grad = False
```

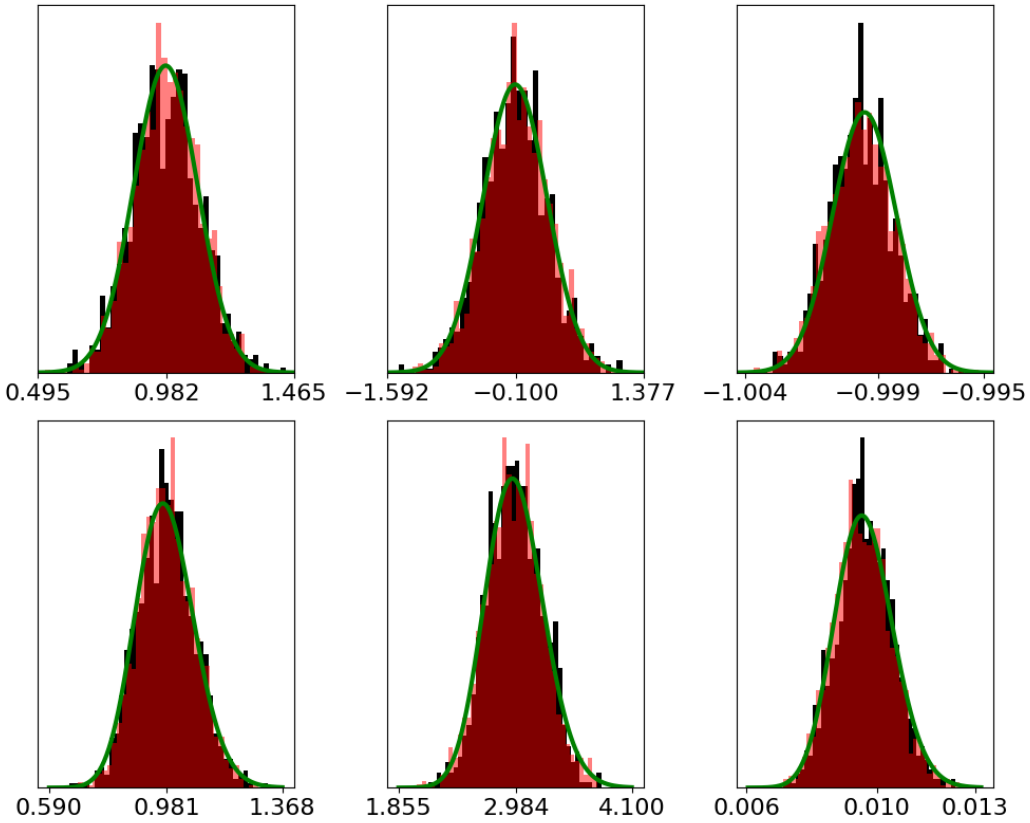

Figure 7: Distributions of $\mu^{\ell,k}$ (top row) and $\sigma^{\ell,k}$ (bottom row) which are the BN centering and scaling statistics computed at each mini-batch for three units in a DN layer with $\boldsymbol{W}^{\ell}\boldsymbol{z}^{\ell-1} \sim \mathcal{N}([1, 0, -1], \text{diag}([1, 3, 0.1]))$ (ℓ is not relevant in this context). The empirical distributions in **black** are obtained by repeatedly sampling mini-batches of size 64 from a training set of size 1000. The analytical distributions in **green** can be obtained analysis, and its empirical distributions in **red** closely match the training one. However by doing the amount of noise can be controlled and is now independent of the mini-batch size.

E EXPERIMENTAL DETAILS

E.1 DETAILS ON IMAGENET EXPERIMENTS

Architectures and hyper-parameters. We use a Resnet50 bottleneck, optimized during 100 epochs with LARS and an initial learning rate of 0.3 for all methods. The batch size is 1024 for all methods except SimCLR. All methods have projector $8192 - 8192 - 8192$ except DINO and Supervised. Throughout the training, we evaluate the learned representation using an online linear classifier. Details specific to each method:

- Barlow Twins: for Experiment 5.1, we have off-diagonal coefficient 0.0051.
- VICReg: for Experiment 5.1, we have projector size $8192 - 8192 - 8192$, Invariance coeff 25, Variance coeff 25 and Covariance coeff 1. For the rest of the experiments, and following Bardes et al. (2022), we only move the covariance coefficient when changing the size of the projector. We scale it in the square root of the output size of the projector in order to have similar magnitude for the covariance term for all projector sizes.
- SimCLR: for Experiment 5.1, we choose a higher batch size of 2048 as SimCLR is sensitive to this parameter (Chen et al., 2020). The temperature is 0.15.
- DINO: we use 8 crops. The head has 4 layers of widths $2048 - 2048 - 2048 - 256 - 65536$.

Augmentations. At train time, the resolution is 160 and we use

- `RandomHorizontalFlip()`.
- `ColorJitter(0.8, 0.4, 0.4, 0.2, 0.1)`.
- `Greyscale(0.2)`.
- `NormalizeImage` with ImageNet mean and standard deviation.
- `GaussianBlur()` with kernel size $(5, 9)$ and sigma $(0.1, 2)$.

At validation time, the resolution is 224 and the images are cropped and normalized with ImageNet mean and standard deviation.

E.2 ICA SETUP

Detailed setup. In these experiments, we keep the same optimizer, train on 100 epochs and choose a batch size of 64 for both datasets. We tune the learning rate according to a logarithmic grid $[1.0, 10.0, 100.0]$ (then rescaled according to $\frac{lr \times batchsize}{256}$). The MLP can be either a SSL-like projector or simply a fully-connected layer followed by a ReLU. The wider the MLP, the better the result hence it does not require selection. For the synthetic dataset, the MLP has width 1024, and 8192 for the audio one.

- Linear ICA: We tune the standard and covariance coefficients according to a logarithmic grid $[1, 10, 100]$.
- PNL ICA: Our architecture for the nonlinear ICA experiments is presented in Figure 8. The encoder f is a MLP with 3 layers and width 128. The decoder h is an learnable nonlinearity (a MLP with 3 hidden layers and depth 16) followed by a fully-connected layer. We tune the standard, covariance and reconstruction coefficients according to a larger logarithmic grid $[1, 3, 10, 30, 100]$.

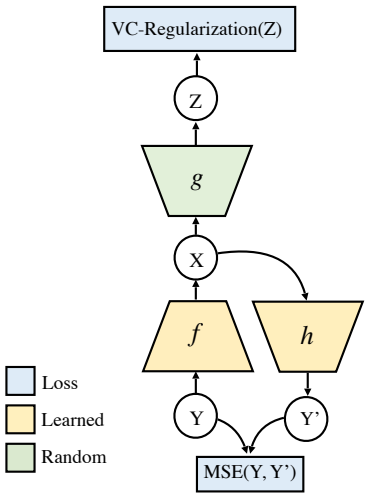

Figure 8: Nonlinear ICA model.

Reconstructed sources for ICA. In this paragraph, we provide ground truth sources for the synthetic data along with examples of reconstructed sources.

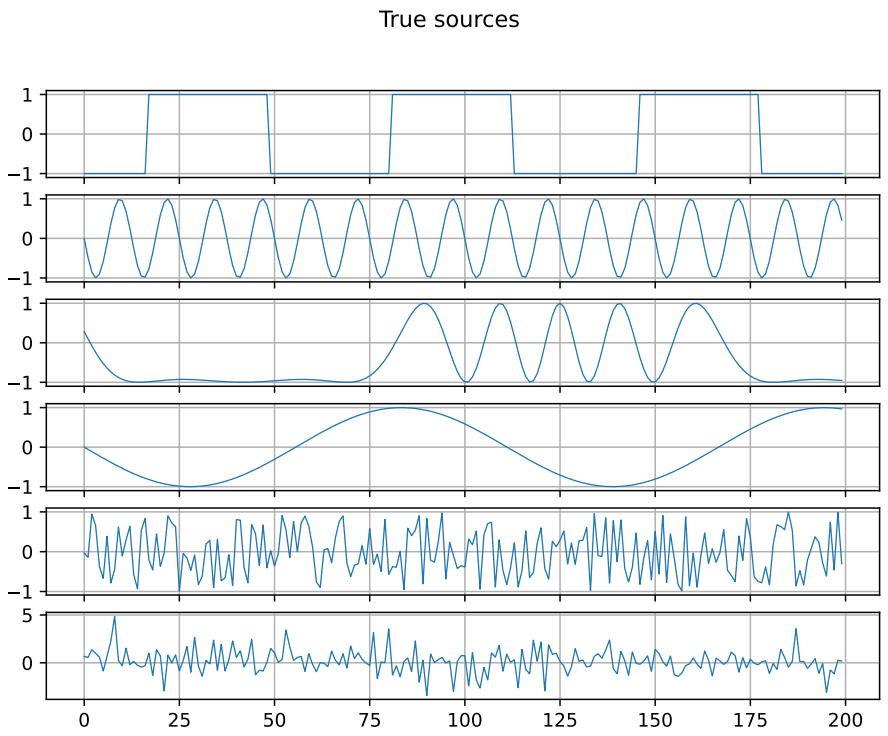

Figure 9: Data before mixing.

Figure 10: Data after mixing, before feeding to the various ICA models.

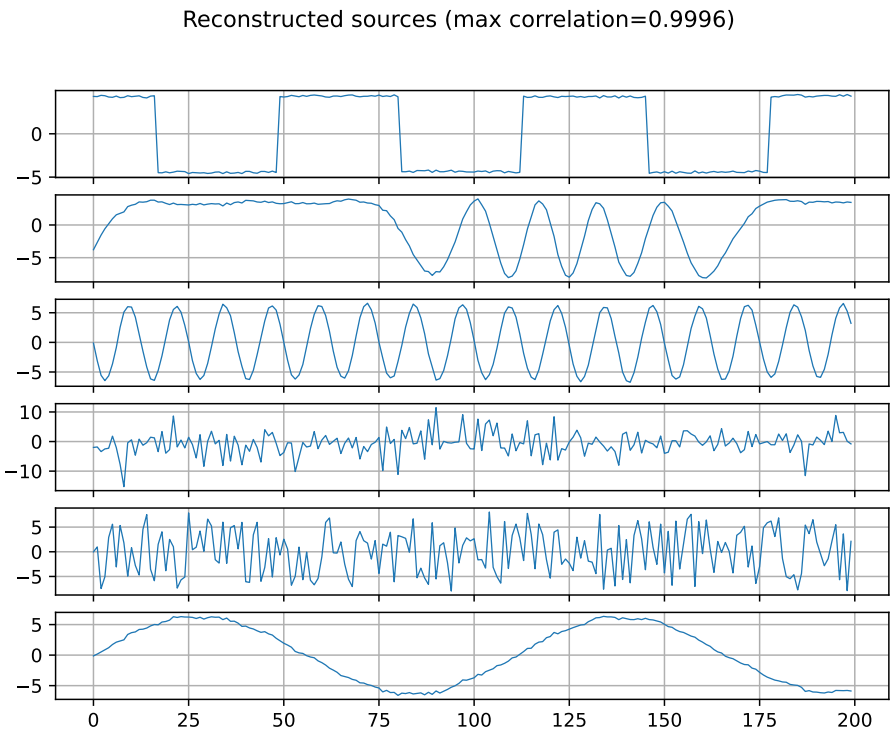

Figure 11: This level of max correlation is typically achieved by FastICA, VCReg or Anica.

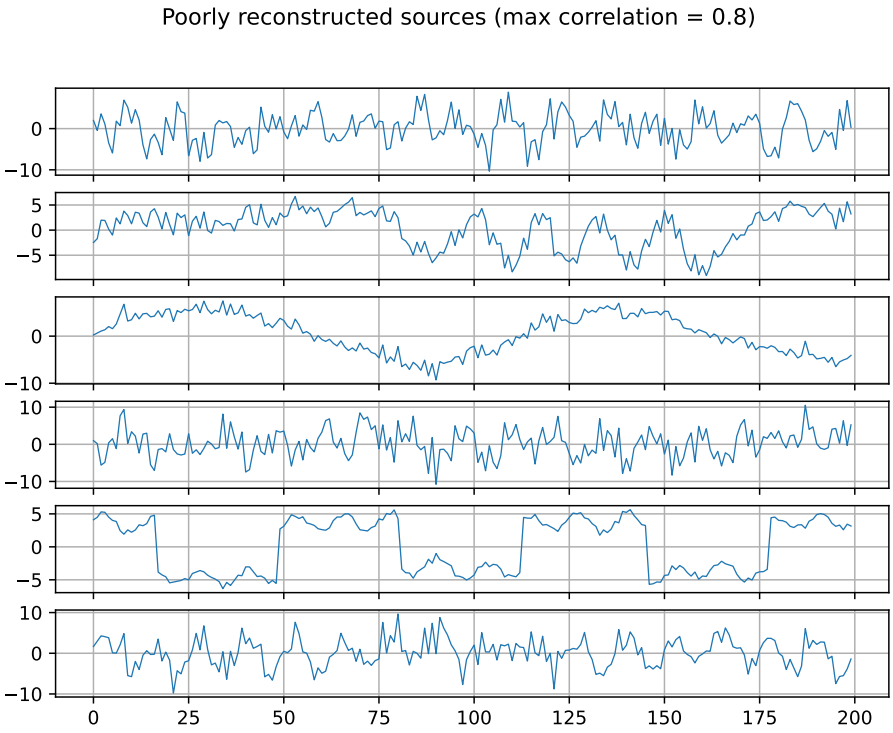

Figure 12: This level of max correlation is typically achieved by Whitening.