

## Article

# Kernel VICReg for Self-Supervised Learning in Reproducing Kernel Hilbert Space

M. Hadi Sepanj<sup>1,\*</sup>, Benjamin Ghogh<sup>2,†</sup>, Saed Moradi<sup>1</sup> and Paul Fieguth<sup>1</sup><sup>1</sup> Vision and Image Processing Group, Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; paul.fieguth@uwaterloo.ca (P.F.)<sup>2</sup> Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; bghogh@uwaterloo.ca

\* Correspondence: mhsepanj@uwaterloo.ca

† These authors contributed equally to this work.

## Abstract

Self-supervised learning (SSL) has emerged as a powerful paradigm for representation learning by optimizing geometric objectives, such as invariance to augmentations, variance preservation, and feature decorrelation, without requiring labels. However, most existing methods operate in Euclidean space, limiting their ability to capture nonlinear dependencies and geometric structures. In this work, we propose Kernel VICReg, a novel self-supervised learning framework that pulls the VICReg objective into a Reproducing Kernel Hilbert Space (RKHS). By kernelizing each term of the loss, variance, invariance, and covariance, we obtain a general formulation that operates on double-centered kernel matrices and Hilbert–Schmidt norms, enabling nonlinear feature learning without explicit mappings. We demonstrate that Kernel VICReg mitigates the risk of representational collapse under challenging conditions and improves performance on datasets exhibiting nonlinear structure or limited sample regimes. Empirical evaluations across MNIST, CIFAR-10, STL-10, TinyImageNet, and ImageNet100 show consistent gains over Euclidean VICReg, with particularly strong improvements on datasets where nonlinear structures are prominent. UMAP visualizations are provided only as a qualitative illustration of embedding geometry and are not used as a calibration or statistical validation. Our results suggest that kernelizing SSL objectives is a promising direction for bridging classical kernel methods with modern representation learning.

**Keywords:** self-supervised learning; VICReg; Reproducing Kernel Hilbert Space

Academic Editor: Francesco Archetti

Received: 28 January 2026

Revised: 26 February 2026

Accepted: 3 March 2026

Published: 5 March 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Self-supervised learning (SSL) has emerged as a dominant paradigm for representation learning by leveraging the underlying structure of data without the need for human-annotated labels [1,2]. Methods such as SimCLR [1], BYOL [3], VICReg [2], and Barlow Twins [4] have demonstrated remarkable performance by enforcing objectives such as invariance to augmentations, feature decorrelation, and variance preservation. However, relying on Euclidean representations in standard self-supervised learning objectives often assumes a relatively simple geometric structure in the latent space. After multiple layers of nonlinear transformation, this assumption becomes questionable, since latent representations are likely to inhabit a highly non-linear manifold, poorly characterized by standard second-order statistics or  $\ell_2$  distances. This motivates our kernelized formulation, which

enables learning in an implicitly defined high-dimensional feature space that captures the underlying manifold structure.

Several recent works have incorporated kernels into SSL objectives, e.g., [5–9]. These methods typically replace similarity metrics or introduce kernel-based dependence criteria within contrastive or predictive frameworks. In contrast, our approach performs a structural lifting of VICReg itself: the variance and covariance penalties are rederived from the covariance operator in RKHS, rather than modified heuristically. This distinction is important, as it preserves the collapse-prevention mechanism of VICReg while redefining its geometry.

Kernel methods, long celebrated for their ability to implicitly map data into high-dimensional feature spaces via the kernel trick [10], offer a compelling avenue to address this limitation. In supervised settings, the transformation from linear to nonlinear models via kernelization (exemplified by the transition from linear SVM to kernel SVM) has been foundational in classical machine learning. Inspired by this paradigm, we ask: *can core SSL losses be systematically lifted into a Reproducing Kernel Hilbert Space?* Here, we answer this by showing how one can replace Euclidean-space objectives with their RKHS counterparts, exemplified through a kernelized version of VICReg.

We propose *Kernel-VICReg*, which computes invariance, variance, and covariance entirely in RKHS via double-centered kernels and Hilbert–Schmidt norms, while we focus on VICReg as a concrete example (and note that Barlow Twins admits an analogous kernelization), the same RKHS-lifting could be extended to contrastive frameworks like SimCLR or predictive ones like BYOL with suitable cross-kernel formulations.

Experiments across multiple datasets show that Kernel-VICReg can yield more effective representations than its Euclidean counterpart under our evaluation protocol and can improve stability in settings where Euclidean VICReg collapses. These results suggest that integrating kernel methods into SSL is a promising direction.

### 1.1. Related Work: Kernel Methods in Self-Supervised Learning

While several recent SSL methods incorporate kernel functions, they fundamentally differ from Kernel VICReg in their application and scope. Existing approaches generally fall into the following categories:

- **Use of Nonlinear Dependence in SSL:** Some methods such as [5,8] use Hilbert–Schmidt Independence Criterion (HSIC) [11] to model nonlinear dependence of samples or features in RKHS.
- **Kernels as Regularizers:** Some methods, such as [9], integrate kernel tools, such as Maximum Mean Discrepancy (MMD) [12], into existing SSL losses to align feature distributions. In these cases, the kernel is used as a supplementary distance metric, but the underlying architecture remains Euclidean.
- **Implicit Kernel Analysis:** Other works, such as [13], use kernels primarily as a theoretical lens to analyze the training dynamics or “spectral bias” of standard SSL objectives like SimCLR [1] or Barlow Twins [4].

In contrast, our proposed Kernel VICReg represents a systematic lifting of an entire loss function of an existing SSL method, i.e., VICReg, into the RKHS. Unlike methods that use kernels for specific terms, we kernelize the variance, invariance, and covariance terms simultaneously. This allows the model to operate on double-centered kernel matrices and Hilbert–Schmidt norms, capturing nonlinear dependencies across the entire loss function rather than just regularizing a single component. To the best of our knowledge, this is the first work to provide a complete kernelized derivation of the VICReg framework.

## 1.2. Positioning and Novelty

Recent works have explored the use of kernels in self-supervised learning, including kernel-based contrastive objectives and kernelized predictive frameworks. However, these methods typically replace similarity measures or dependence criteria within existing objectives (e.g., kernelized contrastive alignment or kernel dependence maximization), rather than systematically lifting the entire regularization structure of a non-contrastive SSL method into RKHS.

Our contribution is structurally different. We show that all three components of VICReg—invariance, variance preservation, and covariance decorrelation—admit principled RKHS counterparts derived from the covariance operator in Hilbert space. This yields a unified formulation in which:

1. invariance is expressed via cross-kernel trace distances,
2. variance preservation is tied to eigenvalues of the centered kernel matrix, and
3. covariance decorrelation becomes a Hilbert–Schmidt norm penalty.

To our knowledge, a full operator-level lifting of VICReg into RKHS using covariance operators and double-centered Gram matrices has not been previously derived. Importantly, our formulation is not a kernel substitution in the similarity function but a redefinition of the geometry in which the SSL objective is defined.

The proposed Kernel VICReg framework provides a more robust geometric constraint in the RKHS, while standard Euclidean methods may suffer from dimensional collapse when the projection head is not sufficiently wide. Our kernelized approach enhances robustness to representation collapse by leveraging the infinite-dimensional nature of the Hilbert space to maintain feature variance.

## 2. Review of VICReg

### 2.1. Data and Network Settings

The neural network is composed of an encoder network and an expander network. Let  $\mathcal{X} := \{x_1, x_2, \dots, x_n\}$  be the dataset where  $x_i \in \mathbb{R}^d$  and  $d$  is the dimensionality of data. Every batch of data, with size  $b$ , contains  $\mathcal{X}_b := \{x_1, x_2, \dots, x_b\}$  and  $\mathcal{X}'_b := \{x'_1, x'_2, \dots, x'_b\}$  where  $x_i$  and  $x'_i$  correspond to each other; e.g., they are augmentations of a common underlying image. They pass through an encoder network to output the  $q$ -dimensional latent embeddings  $\{y_i\}_{i=1}^b$  and  $\{y'_i\}_{i=1}^b$  where  $y_i, y'_i \in \mathbb{R}^q$ . The latent embeddings pass through the expander network to obtain the  $p$ -dimensional output embeddings  $\{z_i\}_{i=1}^b$  and  $\{z'_i\}_{i=1}^b$  where  $z_i, z'_i \in \mathbb{R}^p$  where  $p > q$ .

### 2.2. VICReg Loss Function

VICReg [2] is a non-contrastive SSL method that simultaneously enforces three complementary properties on a pair of views  $x$  and  $x'$  of the same sample:

1. Invariance: embeddings of corresponding views should be close;
2. Variance preservation: each dimension of the embedding space should maintain sufficient spread to avoid collapse;
3. Covariance decorrelation: different embedding dimensions should be de-correlated with encourage richness.

The VICReg loss function contains three terms reflecting the three preceding properties: The *invariance loss* is defined as:

$$\mathcal{L}_{\text{inv}}(x, x') = \frac{1}{b} \sum_{i=1}^b \|z_i - z'_i\|_2^2, \quad (1)$$

which encourages paired views to have similar representations.

The *variance loss* seeks to ensure that no dimension collapses to zero variance, and is defined as

$$\mathcal{L}_{\text{var}}(\mathbf{x}) = \frac{1}{p} \sum_{j=1}^p [\gamma - \sigma_j]_+^2, \quad (2)$$

for threshold  $\gamma > 0$  and where  $[\cdot]_+ = \max(\cdot, 0)$ . The sample standard deviation is found as:

$$\sigma_j = \sqrt{\text{Var}(\{z_{i,j}\}_{i=1}^b)} + \epsilon, \quad (3)$$

where parameter  $\epsilon > 0$  prevents degeneracy, and  $z_{ij}$  denotes the  $j$ -th dimension of  $\mathbf{z}_i$ .

For the *covariance loss*, let

$$\mathbf{C} = 1/(b-1) \tilde{\mathbf{Z}}^\top \tilde{\mathbf{Z}} \in \mathbb{R}^{p \times p}, \quad (4)$$

be the empirical covariance of the zero-centered embeddings  $\tilde{\mathbf{Z}} \in \mathbb{R}^{b \times p}$ . VICReg penalizes off-diagonal correlations:

$$\mathcal{L}_{\text{cov}}(\mathbf{x}) = \frac{1}{p} \sum_{j=1}^p \sum_{k=1, k \neq j}^p \mathbf{C}_{jk}^2, \quad (5)$$

where  $\mathbf{C}_{jk}$  denotes the  $(j, k)$ -th element of the matrix  $\mathbf{C}$ . Promoting a diagonal covariance matrix encourages different features to capture distinct aspects of the data.

The *Overall VICReg Loss* then is a weighted summation of the preceding terms:

$$\begin{aligned} \mathcal{L}_{\text{VICReg}} = & \alpha \mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}') + \beta (\mathcal{L}_{\text{var}}(\mathbf{x}) + \mathcal{L}_{\text{var}}(\mathbf{x}')) \\ & + \zeta (\mathcal{L}_{\text{cov}}(\mathbf{x}) + \mathcal{L}_{\text{cov}}(\mathbf{x}')), \end{aligned} \quad (6)$$

with hyperparameters  $\alpha, \beta, \zeta > 0$ . VICReg thus avoids the need for negative samples or momentum encoders by combining these three regularizers in Euclidean space, laying the groundwork for our kernelized extension in the next section.

### 3. Kernel VICReg

Standard VICReg operates in Euclidean space by enforcing variance preservation, decorrelating feature dimensions, and ensuring consistency between views of the same sample. Our proposed Kernel VICReg extends these principles to the Reproducing Kernel Hilbert Space (RKHS), allowing for non-linear representations without explicit feature mappings. We rigorously derive the kernelized counterparts of VICReg's variance, covariance, and invariance losses.

#### 3.1. Covariance in RKHS

Before deriving the kernelized VICReg loss components, we establish the fundamental result that the covariance operator in RKHS is proportional to the double-centered kernel matrix. The formulations derived and explained in this section will be used in kernelizing the loss terms of VICReg.

Let  $\phi(\mathbf{x})$  be the implicit feature mapping in RKHS  $\mathcal{H}$ , where  $\phi(\cdot)$  is the pulling function into RKHS. The kernel function in the RKHS  $\mathcal{H}$  is defined as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}, \quad (7)$$

where  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  denotes the inner product in the RKHS  $\mathcal{H}$ .

The covariance operator in RKHS is defined as [14,15]:

$$\mathbb{R}^{t \times t} \ni \mathbf{C}_\phi(\mathbf{x}) = \frac{1}{b} \sum_{i=1}^b (\phi(\mathbf{x}_i) - \bar{\phi})(\phi(\mathbf{x}_i) - \bar{\phi})^\top, \quad (8)$$

where  $t$  is the dimensionality of the RKHS,  $\phi(\mathbf{x}_i) - \bar{\phi}$  is the centered feature map of the batch, and  $\bar{\phi} = \frac{1}{b} \sum_{j=1}^b \phi(\mathbf{x}_j)$  is the mean feature vector of the batch in RKHS. If we let

$$\Phi := [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_b)] \in \mathbb{R}^{t \times b}, \quad (9)$$

then Equation (8) can be restated in matrix form:

$$\mathbb{R}^{t \times t} \ni \mathbf{C}_\phi(\mathbf{x}) = \frac{1}{b} (\Phi \mathbf{H})(\Phi \mathbf{H})^\top, \quad (10)$$

where  $\mathbf{H} := \mathbf{I}_b - \frac{1}{b} \mathbf{1}_b \mathbf{1}_b^\top \in \mathbb{R}^{b \times b}$  is the centering matrix.

Using the kernel trick, we define kernel matrix  $\mathbf{K} \in \mathbb{R}^{b \times b}$  whose  $(i, j)$ -th element is:

$$\mathbf{K}(\mathbf{x})[i, j] = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}. \quad (11)$$

From Equation (9), the kernel matrix can be stated as:

$$\mathbf{K}(\mathbf{x}) = \Phi^\top \Phi. \quad (12)$$

The double-centered kernel matrix is:

$$\hat{\mathbf{K}}(\mathbf{x}) := \mathbf{H} \mathbf{K}(\mathbf{x}) \mathbf{H} \stackrel{(12)}{=} \mathbf{H} \Phi^\top \Phi \mathbf{H} \stackrel{(s)}{=} (\Phi \mathbf{H})^\top (\Phi \mathbf{H}), \quad (13)$$

where step (s) considers that the centering matrix  $\mathbf{H}$  is symmetric. The double-centered kernel matrix will be used in the following.

The following explains the relation of the covariance operator and double-centered kernel in RKHS. The squared Hilbert–Schmidt norm of the covariance matrix in RKHS is:

$$\begin{aligned} \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}^2 &= \text{tr}(\mathbf{C}_\phi(\mathbf{x})^\top \mathbf{C}_\phi(\mathbf{x})) \\ &\stackrel{(10)}{=} \frac{1}{b^2} \text{tr}((\Phi \mathbf{H})(\Phi \mathbf{H})^\top (\Phi \mathbf{H})(\Phi \mathbf{H})^\top) \\ &\stackrel{(a)}{=} \frac{1}{b^2} \text{tr}((\Phi \mathbf{H})^\top (\Phi \mathbf{H})(\Phi \mathbf{H})^\top (\Phi \mathbf{H})) \\ &\stackrel{(13)}{=} \frac{1}{b^2} \text{tr}(\hat{\mathbf{K}}(\mathbf{x}) \hat{\mathbf{K}}(\mathbf{x})) \\ &= \frac{1}{b^2} \text{tr}(\hat{\mathbf{K}}(\mathbf{x})^2), \end{aligned} \quad (14)$$

where  $\text{tr}(\cdot)$  denotes the trace of matrix and step (a) follows from the cyclic property of the trace operator. For easier computation in computer, it is possible to restate Equation (14) using the Frobenius norm:

$$\begin{aligned} \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}^2 &= \frac{1}{b^2} \text{tr}(\hat{\mathbf{K}}(\mathbf{x})^2) = \frac{1}{b^2} \sum_{i \neq j} [\hat{\mathbf{K}}(\mathbf{x})]_{i,j}^2 \\ &= \frac{1}{b^2} \left( \|\hat{\mathbf{K}}(\mathbf{x})\|_F^2 - \sum_{i=1}^b [\hat{\mathbf{K}}(\mathbf{x})]_{i,i}^2 \right). \end{aligned} \quad (15)$$

From Equation (14), the covariance operator in RKHS is proportional to the centered kernel matrix:

$$\mathbf{C}_\phi(\mathbf{x}) \propto \frac{1}{b} \hat{\mathbf{K}}(\mathbf{x}). \quad (16)$$

This key result enables the kernelization of VICReg's variance and covariance regularization terms, as discussed next.

### 3.2. Kernelized Variance Regularization

The variance regularization term in VICReg prevents representation collapse by ensuring that the variance along each feature dimension remains sufficiently large, i.e., above a threshold  $\gamma$ . In RKHS, the variance of feature dimensions corresponds to the eigenvalues of  $\mathbf{C}_\phi(\mathbf{x})$ . Since  $\mathbf{C}_\phi(\mathbf{x})$  is proportional to  $\hat{\mathbf{K}}(\mathbf{x})$  according to Equation (16), we define the kernelized variance loss as:

$$\mathcal{L}_{\text{var}}(\mathbf{x}) = \frac{1}{b} \sum_{i=1}^b \left( \left[ \gamma - \sqrt{\frac{\lambda_i}{b} + \epsilon} \right]_+ \right)^2, \quad (17)$$

where  $[\cdot]_+ := \max(\cdot, 0)$  is the standard Hinge loss,  $\{\lambda_i\}_{i=1}^b$  are the eigenvalues of  $\hat{\mathbf{K}}(\mathbf{x})$ ,  $\gamma$  is a threshold for the minimum desired standard deviation, and  $\epsilon$  is a small positive number preventing numerical instabilities. The proof of why  $\lambda_i/b$  is understood as a variance in RKHS will be developed in Section 4.1.

It is noteworthy that computing the eigenvalues here is not concerning in terms of time of computation because the double-centered kernel is  $b \times b$  where  $b$  is the batch size, which is usually not a very large number.

### 3.3. Kernelized Covariance Regularization

To prevent redundancy in representations, VICReg penalizes off-diagonal elements of the covariance matrix (see Equation (5)). Building on Equation (15), the kernelized covariance loss can be defined as

$$\begin{aligned} \mathcal{L}_{\text{cov}}(\mathbf{x}) &= \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}} \\ &= \frac{1}{b} \sqrt{\left( \|\hat{\mathbf{K}}(\mathbf{x})\|_F^2 - \sum_{i=1}^b [\hat{\mathbf{K}}(\mathbf{x})]_{i,i}^2 \right)}. \end{aligned} \quad (18)$$

Because of the direct relation between covariance and correlation, this regularization enforces decorrelation between features in RKHS.

The choice of using the Hilbert–Schmidt norm  $\|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}$  in Equation (18) rather than its square  $\|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}^2$  (as initially suggested by the expansion in Equation (15)) is a deliberate design choice aimed at improving optimization stability. Mathematically, if  $\lambda_i$  are the singular values of the covariance operator, the squared norm  $\|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}^2 = \sum \lambda_i^2$  penalizes large correlations quadratically, which can lead to vanishing gradients for smaller correlation values during the late stages of training. By using the square root form (the norm itself), the gradient magnitude remains more consistent:

$$\frac{\partial \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}}{\partial \theta} = \frac{1}{2 \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}} \frac{\partial \|\mathbf{C}_\phi(\mathbf{x})\|_{\text{HS}}^2}{\partial \theta}. \quad (19)$$

Empirically, we observed that this formulation provides a more balanced optimization landscape, preventing the covariance term from being dominated by a few large off-diagonal correlations and ensuring that all dimensions of the RKHS embedding are decorrelated effectively. This is consistent with recent findings in SSL literature suggesting that normal-

ization of loss terms can lead to smoother convergence and better numerical stability in high-dimensional feature spaces.

### 3.4. Kernelized Invariance Term

The invariance term of the loss function minimizes the mean squared error between corresponding samples  $\mathbf{x}$  and  $\mathbf{x}'$ , i.e., different views of the same sample. Consider the following  $(b \times b)$  kernel matrices:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x})[i, j] &= k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}, \\ K(\mathbf{x}', \mathbf{x}')[i, j] &= k(\mathbf{x}'_i, \mathbf{x}'_j) = \langle \phi(\mathbf{x}'_i), \phi(\mathbf{x}'_j) \rangle_{\mathcal{H}}, \\ K(\mathbf{x}, \mathbf{x}')[i, j] &= k(\mathbf{x}_i, \mathbf{x}'_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}'_j) \rangle_{\mathcal{H}}. \end{aligned}$$

Given kernel matrices  $K(\mathbf{x}, \mathbf{x})$  and  $K(\mathbf{x}', \mathbf{x}')$  for two augmented views and their cross-kernel matrix  $K(\mathbf{x}, \mathbf{x}')$ , the distance of the views in RKHS is defined as [16]:

$$\mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}') = \frac{1}{b} \text{tr}(K(\mathbf{x}, \mathbf{x}) + K(\mathbf{x}', \mathbf{x}') - 2K(\mathbf{x}, \mathbf{x}')). \quad (20)$$

This loss term pushes the corresponding, i.e., augmented, instances toward each other in the RKHS and pulls away the non-corresponding instances away from each other in the RKHS. This enforces consistency across augmentations in RKHS.

### 3.5. Overall Kernel VICReg Loss

Combining all three regularization terms, the final Kernel VICReg loss is given by:

$$\begin{aligned} \mathcal{L}_{\text{Kernel-VICReg}} &= \alpha \mathcal{L}_{\text{inv}}(\mathbf{x}, \mathbf{x}') + \beta (\mathcal{L}_{\text{var}}(\mathbf{x}) + \mathcal{L}_{\text{var}}(\mathbf{x}')) \\ &\quad + \zeta (\mathcal{L}_{\text{cov}}(\mathbf{x}) + \mathcal{L}_{\text{cov}}(\mathbf{x}')), \end{aligned} \quad (21)$$

where  $\alpha, \beta, \zeta$  are hyperparameters controlling the contributions of invariance, variance, and covariance. Note that the best values for hyperparameters  $\alpha, \beta, \zeta, \gamma, \epsilon$  differ across the VICReg and Kernel VICReg. The best hyperparameters can be found depending on the dataset, as they are found in the original VICReg.

By reformulating VICReg in RKHS, our method enables self-supervised learning in high-dimensional implicit feature spaces without explicit feature extraction, making it a powerful framework for non-linear representation learning.

## 4. Discussions

### 4.1. Relation of Kernelized Variance Term with Kernel PCA

There is a close relation between the kernelized variance term in Kernel VICReg and kernel Principal Component Analysis (kernel PCA) [14]. In standard PCA, the eigenvalues of the covariance matrix  $\mathbf{C} \in \mathbb{R}^{p \times p}$  are calculated using the following eigenvalue problem:

$$\mathbf{C} \mathbf{u}_i = \eta_i \mathbf{u}_i, \quad (22)$$

where  $\eta_i \in \mathbb{R}$  and  $\mathbf{u}_i \in \mathbb{R}^p$  are the  $i$ -th eigenvalue and eigenvector of the covariance matrix  $\mathbf{C}$ , respectively.

In kernel PCA, the eigenvalue problem of the double-centered kernel matrix  $\hat{\mathbf{K}}(\mathbf{x}) \in \mathbb{R}^{b \times b}$  is considered:

$$\hat{\mathbf{K}}(\mathbf{x}) \mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad (23)$$

where  $\lambda_i \in \mathbb{R}$  and  $\mathbf{v}_i \in \mathbb{R}^b$  are the  $i$ -th eigenvalue and eigenvector of the double-centered kernel matrix  $\hat{\mathbf{K}}(x)$ , respectively.

Each eigenvector  $\mathbf{v}_i$  gives a principal direction in the feature space  $\phi(\mathbf{U}) \in \mathbb{R}^{t \times b}$ . According to the representation theory, any function in RKHS lies in the span of all points in the RKHS [17]:

$$\phi(\mathbf{U}) = \sum_{i=1}^b \alpha_i \phi(x_i) = \Phi \mathbf{A}, \quad (24)$$

where  $\Phi \in \mathbb{R}^{t \times b}$  is defined in Equation (9) and  $\mathbf{A} := [\alpha_1, \dots, \alpha_b]^\top \in \mathbb{R}^{b \times b}$  is the matrix of coefficients in the linear combination.

On the one hand, the variance of the principal direction in the feature space is:

$$\begin{aligned} \text{Var}(\phi(\mathbf{U})) &= \frac{1}{b} \|(\Phi \mathbf{H})^\top \mathbf{A}\|_2^2 \\ &= \frac{1}{b} \text{tr}(((\Phi \mathbf{H})^\top \mathbf{A})^\top ((\Phi \mathbf{H})^\top \mathbf{A})) \\ &= \frac{1}{b} \text{tr}(\mathbf{A}^\top \Phi \Phi^\top \mathbf{A}) \\ &\stackrel{(24)}{=} \frac{1}{b} \text{tr}(\mathbf{A}^\top \underbrace{(\Phi \mathbf{H})^\top (\Phi \mathbf{H})}_{\hat{\mathbf{K}}(x)} \underbrace{(\Phi \mathbf{H})^\top (\Phi \mathbf{H})}_{\hat{\mathbf{K}}(x)} \mathbf{A}) \\ &= \frac{1}{b} \text{tr}(\mathbf{A}^\top \hat{\mathbf{K}}(x)^2 \mathbf{A}). \end{aligned}$$

For one of the coefficient vectors, this equation becomes:

$$\text{Var}(\phi(u_i)) = \frac{1}{b} \alpha_i^\top \hat{\mathbf{K}}(x)^2 \alpha_i, \quad (25)$$

where trace is dropped because the trace of a scalar is equal to itself.

Assume that the coefficient  $\alpha_i$  is the  $i$ -th eigenvector of the double-centered kernel matrix. Thus, the variance becomes:

$$\text{Var}(\phi(u_i)) = \frac{1}{b} \mathbf{v}_i^\top \hat{\mathbf{K}}(x)^2 \mathbf{v}_i. \quad (26)$$

Squaring the double-centered kernel matrix in Equation (23) gives:

$$\hat{\mathbf{K}}(x)^2 \mathbf{v}_i = \lambda_i^2 \mathbf{v}_i. \quad (27)$$

Substituting Equation (27) in Equation (26) provides:

$$\begin{aligned} \text{Var}(\phi(u_i)) &= \frac{1}{b} \alpha_i^\top \lambda_i^2 \alpha_i = \frac{1}{b} \lambda_i^2 \alpha_i^\top \alpha_i \\ &= \frac{1}{b} \lambda_i^2 \|\alpha_i\|_2^2. \end{aligned} \quad (28)$$

The coefficient is usually normalized to have  $\lambda_i \|\alpha_i\|_2^2 = 1$  (so that  $\|\phi(u_i)\|_{\mathcal{H}} = 1$ ):

$$\alpha_i = \frac{\mathbf{v}_i}{\sqrt{\lambda_i}} \implies \|\alpha_i\|_2^2 = \frac{1}{\lambda_i}, \quad (29)$$

where the eigenvector  $\mathbf{v}_i$  is assumed to be normalized to have unit length, i.e.,  $\|\mathbf{v}_i\|_2 = 1$ . According to Equation (29), Equation (28) becomes:

$$\text{Var}(\phi(u_i)) = \frac{\lambda_i}{b}. \quad (30)$$

This proves why  $\lambda_i/b$  is used as variance in Equation (17).

On the other hand, left-multiplying Equation (23) by  $\mathbf{v}_i^\top$  gives:

$$\mathbf{v}_i^\top \hat{\mathbf{K}}(\mathbf{x}) \mathbf{v}_i = \mathbf{v}_i^\top \lambda_i \mathbf{v}_i = \lambda_i \mathbf{v}_i^\top \mathbf{v}_i = \lambda_i \|\mathbf{v}_i\|_2^2 \stackrel{(a)}{=} \lambda_i,$$

where (a) assumes that the eigenvector  $\mathbf{v}_i$  is normalized to have unit length. Therefore, if  $\|\mathbf{v}_i\|_2 = 1$ , then:

$$\lambda_i = \mathbf{v}_i^\top \hat{\mathbf{K}}(\mathbf{x}) \mathbf{v}_i. \quad (31)$$

Putting Equation (31) in Equation (30) provides:

$$\mathbf{Var}(\phi(\mathbf{u}_i)) = \frac{1}{b} \mathbf{v}_i^\top \hat{\mathbf{K}}(\mathbf{x}) \mathbf{v}_i. \quad (32)$$

This analysis shows that the kernelized variance regularization in the proposed Kernel VICReg can be considered as kernel PCA. A similar analysis can be discussed to analyze the variance regularization term in VICReg as PCA.

#### 4.2. Connection to HSIC and Independence

The squared Hilbert–Schmidt norm of the RKHS covariance operator (used in our covariance loss) is closely related to the Hilbert–Schmidt Independence Criterion (HSIC), a well-established kernel-based dependence measure. In this view, our covariance loss can be interpreted as minimizing feature dependence in RKHS, encouraging the learning of diverse and disentangled features. This theoretical grounding strengthens the regularization effect of the Kernel VICReg loss beyond simple decorrelation.

#### 4.3. Kernel Choice as Inductive Bias

Different kernels induce different geometric priors. For example, the RBF kernel emphasizes local smoothness, the Laplacian kernel allows sharper decision boundaries, and the rational quadratic interpolates between them. Our experiments reveal that no single kernel is optimal across all datasets; instead, performance depends on the match between the dataset structure and the kernel-induced geometry. This makes Kernel VICReg not only robust but also adaptable to task-specific data distributions. A compatible extension to reduce sensitivity to a single kernel choice is to use kernel mixtures, e.g.,  $k = \sum_m w_m k_m$  with  $w_m \geq 0$  and  $\sum_m w_m = 1$ , or more broadly multiple kernel learning (MKL). This is orthogonal to our main contribution (lifting VICReg into RKHS), since the objective depends on Gram matrices and can directly operate on a mixture Gram matrix.

#### 4.4. Comparison of Approaches in Kernel VICReg and Graph-Based Embedding

There exist recent works on kernel-graph integration and spectral clustering (e.g., [18,19]) analyzing how kernel methods interact with graph structure learning and spectral objectives. These approaches focus primarily on unsupervised clustering or graph-based embedding construction. In contrast, our work does not construct or learn a graph structure; instead, we lift a non-contrastive self-supervised regularization objective into RKHS. The role of kernels here is not spectral clustering, but redefining variance and covariance operators in an implicit feature space. Our formulation in Kernel VICReg is therefore complementary rather than competitive with kernel–graph hybrids.

#### 4.5. Comparison of Optimization Objective in Kernel VICReg and Variational Inference

Although our formulation pulls the VICReg objective to an RKHS via kernel covariance operators, the resulting loss remains a deterministic, differentiable functional of the network

parameters. Unlike variational self-supervised methods that introduce latent variables and evidence lower bounds (ELBO), our objective does not involve probabilistic latent modeling or variational inference.

Concretely, the kernelized variance and covariance terms are computed through empirical covariance operators constructed from mini-batch embeddings. These operators depend smoothly on the encoder parameters, and gradients are obtained via automatic differentiation.

Therefore, the optimization problem is a standard stochastic minimization of a deterministic loss (21) optimized using stochastic gradient descent. No variational bound, alternating optimization, or EM-style procedure is introduced by the RKHS lifting.

#### 4.6. Theoretical Properties of Kernel VICReg

##### 4.6.1. Non-Collapse in RKHS

**Proposition 1** (Non-Collapse in RKHS). *Let  $\widehat{\mathbf{K}}(\mathbf{x})$  denote the double-centered kernel matrix of a batch. If the kernelized variance regularization enforces:*

$$\sqrt{\frac{\lambda_i}{b}} \geq \gamma > 0, \quad \forall i \in \{1, \dots, b\}, \quad (33)$$

where  $\{\lambda_i\}_{i=1}^b$  are eigenvalues of  $\widehat{\mathbf{K}}(\mathbf{x})$ , then the covariance operator  $\mathbf{C}_\phi(\mathbf{x})$  in RKHS is strictly positive definite on the span of the batch, and representational collapse (i.e., rank-one embedding) is prevented.

**Proof.** From Equation (16), we have  $\mathbf{C}_\phi(\mathbf{x}) \propto (1/b)\widehat{\mathbf{K}}(\mathbf{x})$ . If all eigenvalues satisfy  $\lambda_i > 0$ , then  $\widehat{\mathbf{K}}$  is full rank on the batch span. Thus the covariance operator has a strictly positive spectrum, implying that no direction in RKHS has zero variance. A collapsed representation corresponds to  $\text{rank}(\widehat{\mathbf{K}}) = 1$ , which contradicts the enforced lower bound.  $\square$

**Remark 1.** Proposition 1 demonstrates that Kernel VICReg enforces spectral spread in RKHS, while Euclidean VICReg only enforces coordinate-wise variance. This is a theoretical distinction between VICReg and Kernel KICReg.

##### 4.6.2. Nonlinear Variance Capture in RKHS

**Theorem 1** (Nonlinear Variance Capture in RKHS). *Let  $\mathcal{M} \subset \mathbb{R}^p$  be a compact nonlinear manifold. Assume that  $\mathcal{M}$  is not contained in any proper affine subspace of  $\mathbb{R}^p$ , but its nonlinear structure cannot be captured by second-order Euclidean statistics (i.e., PCA does not linearize  $\mathcal{M}$ ).*

*Let  $k$  be a universal kernel (e.g., Gaussian RBF or Laplacian) with feature map  $\phi : \mathbb{R}^p \rightarrow \mathcal{H}$ . Then:*

1. *The feature map  $\phi$  is injective on  $\mathcal{M}$ .*
2. *The image  $\phi(\mathcal{M})$  lies in a linear subspace of  $\mathcal{H}$  whose covariance operator encodes nonlinear structure of  $\mathcal{M}$ .*
3. *The eigenvalues of the centered kernel matrix correspond to nonlinear principal components of  $\mathcal{M}$  (kernel PCA).*
4. *Therefore, enforcing lower bounds on eigenvalues of  $\widehat{\mathbf{K}}(\mathbf{x})$  preserves nonlinear modes of variation that are invisible to Euclidean covariance regularization.*

**Proof.** See Appendix A for proof.  $\square$

**Remark 2.** Theorem 1 connects Kernel VICReg to kernel PCA theory, as was also discussed in Section 4.1. This theorem does not claim that RKHS variance always strictly dominates Euclidean

variance, but rather that for universal kernels, nonlinear structure becomes linearly representable in feature space, allowing spectral regularization to act on intrinsic manifold directions.

#### 4.6.3. Spectral Stability in Small Batches

**Theorem 2** (Spectral Stability of Centered Kernel Matrices). Let  $\{\mathbf{x}_i\}_{i=1}^b$  be i.i.d. samples drawn from a distribution  $\mathcal{D}$ . Let  $k$  be a bounded kernel satisfying:

$$0 \leq k(\mathbf{x}, \mathbf{x}) \leq \kappa^2, \quad \forall \mathbf{x}. \quad (34)$$

Let  $\mathbf{K}_b$  denote the  $b \times b$  Gram matrix, and let  $\hat{\mathbf{K}}_b = \mathbf{H}\mathbf{K}_b\mathbf{H}$  be its double-centered version, where  $\mathbf{H} = \mathbf{I}_b - (1/b)\mathbf{1}\mathbf{1}^\top$ . Let  $\boldsymbol{\Sigma} := \mathbb{E}[\hat{\mathbf{K}}_b]$  denote the population centered Gram operator restricted to the sample span. Then, for any  $\delta \in (0, 1)$ , with high probability at least  $1 - \delta$ , we have:

$$\|\hat{\mathbf{K}}_b - \boldsymbol{\Sigma}\|_2 \leq c\kappa^2 \left( \sqrt{\frac{\log(2b/\delta)}{b}} + \frac{\log(2b/\delta)}{b} \right), \quad (35)$$

for some universal constant  $c > 0$ .

**Proof.** See Appendix B for proof.  $\square$

The bound (35) shows that eigenvalue estimates in RKHS concentrate at rate  $\mathcal{O}(1/\sqrt{b})$ , providing stability guarantees for small batch regimes.

**Corollary 1.** Under the above conditions, each eigenvalue  $\lambda_i$  of  $\hat{\mathbf{K}}_b$  concentrates around its population counterpart at rate  $\mathcal{O}(1/\sqrt{b})$ . Therefore, it provides stability guarantees for small-batch regimes where  $b$  is not too large.

#### 4.7. Scalability and Large-Scale Approximations

A primary concern in kernel-based methods is the computational complexity associated with the Gram (kernel) matrix. In Kernel VICReg, the construction of the kernel matrix  $\mathbf{K} \in \mathbb{R}^{b \times b}$  and the eigenvalue decomposition required for the variance loss incur complexities of  $\mathcal{O}(b^2)$  and  $\mathcal{O}(b^3)$ , respectively, where  $b$  is the batch size, while modern hardware efficiently handles standard batch sizes (e.g.,  $b = 256$  to  $2048$ ), scaling to “cognitive computing” levels with massive datasets requires approximation strategies.

##### 4.7.1. Scalability by Nyström Method

To address this, Kernel VICReg can be integrated with the Nyström method [20,21], which approximates the full kernel matrix using  $m$  landmark points ( $m \ll b$ ):

$$\tilde{\mathbf{K}} := \mathbf{K}_{b,m} \mathbf{K}_{m,m}^{-1} \mathbf{K}_{m,b}, \quad (36)$$

where  $\mathbf{K}_{b,m}$  is the matrix of kernel evaluations between the batch samples and the landmarks. This reduces the complexity to  $\mathcal{O}(bm^2)$ .

##### 4.7.2. Scalability by Random Fourier Features

Alternatively, Random Fourier Features (RFF) [22] can be employed for shift-invariant kernels (e.g., RBF kernel). RFF maps the embeddings  $\mathbf{z}$  to a low-dimensional randomized feature space  $\Phi(\mathbf{z}) \in \mathbb{R}^D$  such that the kernel is approximated by a linear inner product:

$$k(\mathbf{z}_i, \mathbf{z}_j) \approx \Phi(\mathbf{z}_i)^\top \Phi(\mathbf{z}_j),$$

$$\Phi(\mathbf{z}) = \sqrt{\frac{2}{D}} \left[ \cos(\boldsymbol{\omega}_1^\top \mathbf{z} + \beta_1), \dots, \cos(\boldsymbol{\omega}_D^\top \mathbf{z} + \beta_D) \right]^\top, \quad (37)$$

where  $\{\omega_i\}_{i=1}^D$  are sampled from the kernel's spectral density. By utilizing RFF, the Kernel VICReg objective simplifies to a linear form in the  $\Phi$ -space, achieving  $\mathcal{O}(bD)$  complexity. This ensures linear scalability relative to the batch size, making the framework suitable for large-scale distributed applications.

#### 4.8. Ethical Considerations and Potential Biases

While Kernel VICReg provides a robust framework for nonlinear representation learning, it is essential to consider the ethical implications regarding data bias. Kernel methods are inherently sensitive to the distribution of the training data. In domains such as medical imaging or cognitive computing, if a specific group is underrepresented, the kernel matrix  $K$  may fail to capture the local geometry of that sub-population, potentially leading to “feature exclusion”, where the RKHS mapping reinforces existing biases.

Furthermore, the choice of kernel (e.g., the RBF kernel width  $\sigma$ ) acts as a scale-dependent filter. If the data from minority groups exhibits different variance scales, a global kernel parameter might sub-optimally encode their features compared to the majority group. To mitigate these risks in sensitive applications, we suggest the use of adaptive kernels or group-fairness constraints within the variance-covariance terms, ensuring that the geometric embedding remains equitable across diverse demographic groups.

## 5. Experimental Results

We evaluate Kernel VICReg on a range of benchmark datasets to assess its ability to learn rich, non-linear representations in self-supervised settings. Our experiments span small-scale datasets (MNIST, CIFAR-10), mid-scale transfer learning (STL-10), and large-scale benchmarks (TinyImageNet, ImageNet100). For all experiments, we use a ResNet-18 backbone as the encoder, followed by a two-layer MLP projector. The model is trained using the Adam optimizer with an initial learning rate of  $3 \times 10^{-4}$  batch size of 256, and cosine learning rate scheduling. Each dataset is augmented following standard protocols: random cropping, horizontal flipping, and color jitter for natural image datasets.

To investigate the effect of different kernels in the Reproducing Kernel Hilbert Space (RKHS), we implement Kernel VICReg using four kernels: linear, radial basis function (RBF), Laplacian, and rational quadratic (RQ). The kernel matrices are computed batch-wise, with double-centering applied to ensure zero-mean embeddings in RKHS. We evaluate the learned representations using linear probing, where a logistic regression classifier is trained atop frozen embeddings, and transfer learning, where encoders pretrained on CIFAR-10 are evaluated on STL-10.

### 5.1. Comparison with Baselines

We compare Kernel VICReg against a diverse set of prominent self-supervised learning methods, including contrastive (SimCLR, MoCo), clustering-based (SwAV), and non-contrastive frameworks (BYOL, DINO, SimSiam, Barlow Twins, and VICReg). All baselines are implemented with the same ResNet-18 encoder to ensure fairness, and we reuse reported numbers or reproduce them when necessary using identical augmentation and optimization settings.

While the field of self-supervised learning (SSL) is rapidly evolving with newer non-contrastive architectures, the primary objective of this work is to provide a theoretical and methodological framework for lifting the VICReg objective into the Reproducing Kernel Hilbert Space (RKHS). Our evaluations focus on comparing Kernel VICReg against its direct Euclidean counterpart and established foundational SSL baselines (e.g., SimCLR, Barlow Twins, and VICReg) to isolate the impact of kernelization. By demonstrating consistent

improvements over the original VICReg across multiple datasets, we establish the validity of the proposed kernelized variance-invariance-covariance constraints.

Table 1 summarizes the top-1 linear evaluation accuracy on ImageNet100 and TinyImageNet. While VICReg exhibits competitive performance on ImageNet100, it collapses on TinyImageNet due to its sensitivity to small datasets with high intra-class variance. In contrast, Kernel VICReg remains stable across all settings, with the Laplacian and RQ kernels achieving the best performance, demonstrating the robustness of RKHS-based regularization.

**Table 1.** Linear evaluation on ImageNet100 and TinyImageNet with ResNet-18 backbone. The performances of baseline methods on TinyImageNet are adapted from [23].

| Model                          | ImageNet100 | TinyImageNet |
|--------------------------------|-------------|--------------|
| SimCLR                         | 78.64       | 37.83        |
| SwAV                           | 74.36       | 35.39        |
| MoCo                           | 79.62       | 41.23        |
| BYOL                           | 80.88       | 36.31        |
| DINO                           | 75.41       | 35.77        |
| SimSiam                        | 78.80       | 27.96        |
| VICReg                         | 79.77       | Collapse     |
| Barlow Twins                   | 80.63       | -            |
| Kernel VICReg—Linear (ours)    | 77.34       | 37.48        |
| Kernel VICReg—RBF (ours)       | 78.14       | 38.21        |
| Kernel VICReg—Laplacian (ours) | 79.92       | 40.12        |
| Kernel VICReg—RQ (ours)        | 79.80       | 40.38        |

Table 2 presents results on MNIST and CIFAR-10. Kernel VICReg consistently outperforms its Euclidean counterpart, particularly on MNIST, where the Laplacian kernel reaches 98.50% accuracy. On CIFAR-10, the RQ kernel achieves the best performance at 86.18%, indicating that kernel choice adapts to data complexity.

**Table 2.** Linear evaluation on MNIST and CIFAR-10 with ResNet-18 backbone.

| Model                   | MNIST | CIFAR-10 |
|-------------------------|-------|----------|
| VICReg                  | 97.15 | 83.41    |
| Kernel VICReg—Linear    | 98.33 | 86.08    |
| Kernel VICReg—RBF       | 92.68 | 81.13    |
| Kernel VICReg—Laplacian | 98.50 | 84.56    |
| Kernel VICReg - RQ      | 97.46 | 86.18    |

Finally, Table 3 reports transfer learning results on STL-10 using encoders pretrained on CIFAR-10. Kernel VICReg transfers better than VICReg, highlighting its generalization capabilities in low-label regimes.

Note that while several variations and incremental improvements to the VICReg architecture have been proposed since its inception in 2022, this study focuses on the foundational task of extending the core VICReg objective into the RKHS. By comparing our method directly against the standard Euclidean VICReg in Tables 2 and 3, we isolate the performance gains attributable to the kernelized covariance and variance constraints. This controlled comparison is essential for validating the theoretical derivation of Kernel VICReg.

**Table 3.** Transfer learning accuracy on STL-10. Embeddings are trained on CIFAR-10 (ResNet-18).

| Model                   | STL-10 Accuracy |
|-------------------------|-----------------|
| VICReg                  | 69.82           |
| Kernel VICReg—Linear    | 71.38           |
| Kernel VICReg—RBF       | 67.21           |
| Kernel VICReg—Laplacian | 71.09           |
| Kernel VICReg—RQ        | 72.34           |

### 5.2. Further Analysis and Insights

To better understand how kernelization affects the structure of learned representations, we visualize the embedding spaces on the MNIST dataset using UMAP for three models: original VICReg, Kernel VICReg with a linear kernel, and Kernel VICReg with a Laplacian kernel (see Figure 1). The UMAP plots reveal differences in cluster geometry across these methods. Representations from standard VICReg exhibit some class separation, but the clusters are elongated and lack compactness, suggesting anisotropic variance and potential feature collapse. The red cluster in VICReg is separated and is constructing two clusters; however, this is not the case for our method.

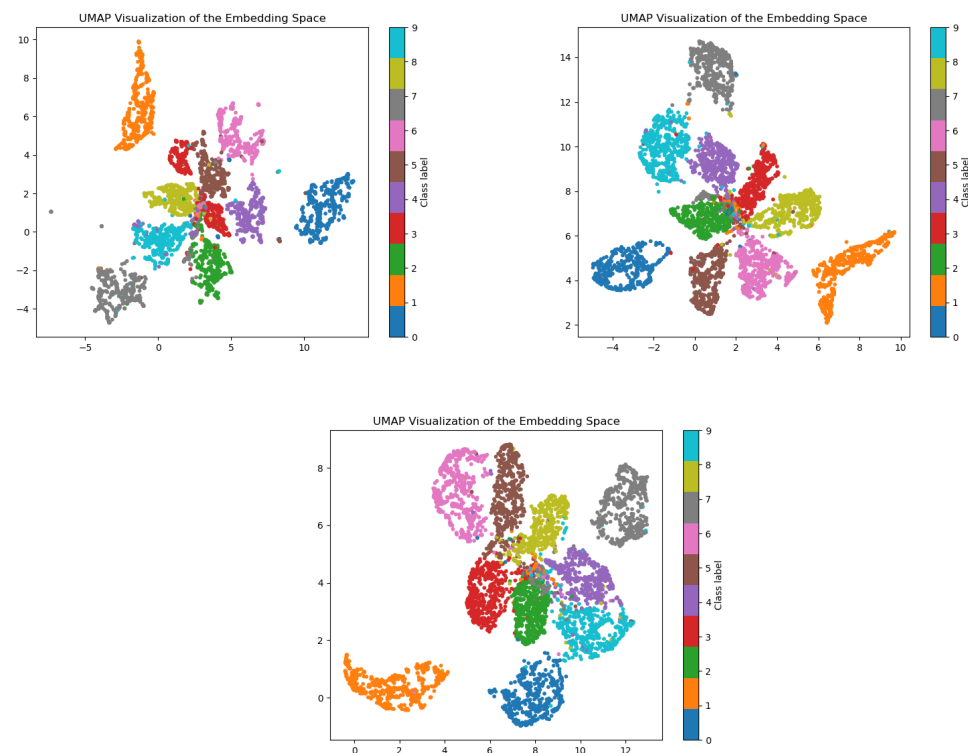


**Figure 1.** UMAP projections (axes: UMAP-1 and UMAP-2) of MNIST embeddings from VICReg (**left**), Kernel VICReg with linear kernel (**middle**), and Kernel VICReg with Laplacian kernel (**right**). Colors denote digit classes (label indices 0–9). The Laplacian kernel yields rounder, more isometric clusters, indicating improved class separability.

Kernel VICReg with a linear kernel improves upon this, producing tighter and more separated clusters, indicating that even without explicit nonlinearity in the kernel, RKHS-based decorrelation provides better structure. However, the most striking improvement appears with the Laplacian kernel: clusters become nearly circular and uniformly spaced, exhibiting strong isometry. This implies that the Laplacian-induced RKHS preserves

pairwise relations and local structure more effectively, leading to embeddings with more consistent intra-class variance and improved inter-class margins.

These visualizations are qualitative; quantitative comparisons are provided by the linear-probe and transfer results in Tables 1 and 3. As shown earlier in Table 1, Kernel VICReg maintains stable and competitive accuracy across both large-scale (ImageNet100) and small-scale (TinyImageNet) benchmarks. Notably, VICReg collapses on TinyImageNet, consistent with its known sensitivity to datasets exhibiting high intra-class variance or insufficient regularization. In contrast, kernelized versions (especially with Laplacian and rational quadratic kernels) perform robustly, demonstrating the benefits of nonlinear geometric alignment in RKHS.

We further analyze robustness sensitivity to kernel hyperparameters. Figure 2 shows that polynomial kernels can be highly sensitive to  $(d, c_0)$  under distribution shifts, while Figure 3 demonstrates that RBF performance is non-monotonic in the bandwidth parameter  $\gamma$ .

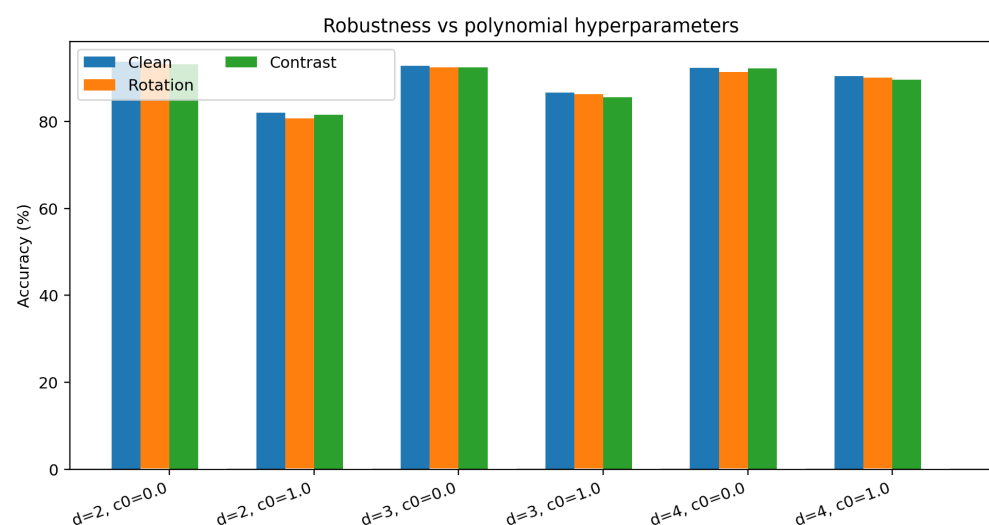


**Figure 2.** Robustness of polynomial-kernel Kernel-VICReg across hyperparameter settings and distribution shifts on MNIST. Each grouped bar corresponds to one  $(d, c_0)$  configuration, where  $d$  is polynomial degree and  $c_0$  is the additive constant in  $k(x, y) = (\gamma x^\top y + c_0)^d$ . Bars report linear-probe accuracy on clean data and under rotation and contrast shift. The spread across groups indicates strong hyperparameter sensitivity, with certain settings preserving clean/shifted performance.

Furthermore, the results in Table 3 highlight the generalization strength of kernel-based methods in transfer learning. Embeddings trained on CIFAR-10 and evaluated on STL-10 show that the RQ and Laplacian kernels outperform VICReg and even the linear kernel variant. These findings support the hypothesis that kernel-induced representations better capture underlying data manifolds, resulting in improved performance on downstream tasks with distributional shifts.

Overall, Kernel VICReg offers a principled extension of VICReg that gracefully incorporates nonlinearity through RKHS-based loss formulations. The improved cluster geometry, resilience to collapse, and higher transfer accuracy together suggest that kernelized self-supervision is a promising direction for representation learning beyond Euclidean limitations.

### 5.3. Implementation Details

We implemented Kernel VICReg in PyTorch 2.6.0 with a modular design that allows kernel choice and parameter tuning through command-line flags. The training pipeline consists of three components: (i) a backbone encoder (either ResNet-18 or a simple CNN), (ii) a

multi-layer perceptron projector, and (iii) the proposed kernelized VICReg loss. Two augmented views are generated per sample following standard SSL protocols, and their embeddings are compared via the kernel-based losses.

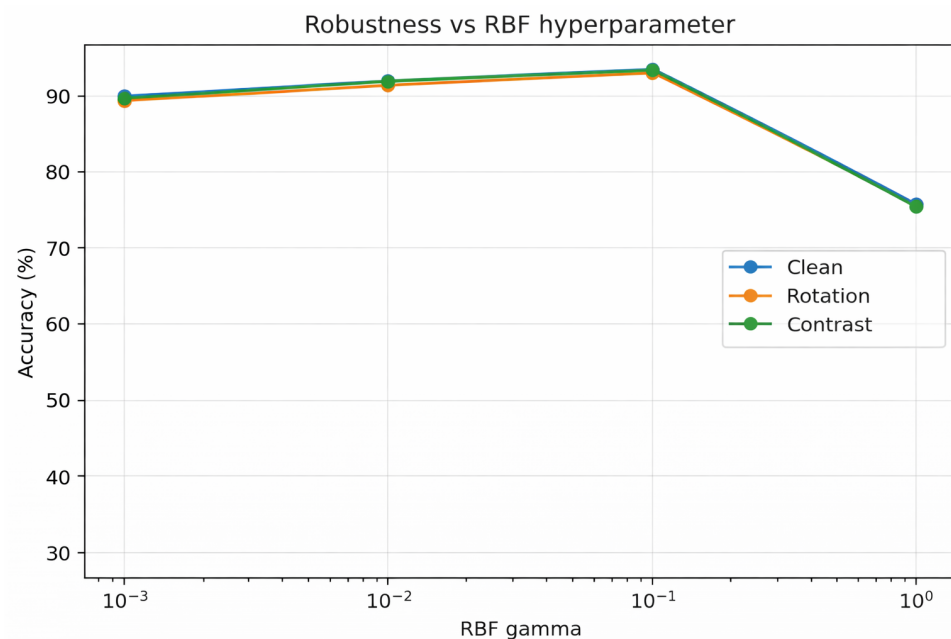


**Figure 3.** Robustness sensitivity of RBF-kernel Kernel-VICReg to the bandwidth parameter  $\gamma$  under clean and shifted MNIST evaluation. Curves report linear-probe accuracy on clean data and under rotation and contrast shift. Performance is non-monotonic with respect to  $\gamma$ : moderate values yield better overall robustness, while overly large  $\gamma$  leads to pronounced degradation, demonstrating the need for careful kernel hyperparameter tuning.

**Kernels and Centering.** We implemented five kernels (linear, polynomial, radial basis function (RBF), Laplacian, and rational quadratic (RQ)) with automatic double-centering to operate in RKHS. For scale-sensitive kernels (RBF, Laplacian, and RQ), the bandwidth parameter  $\gamma$  is adaptively estimated using the median heuristic unless explicitly specified.

**Kernel VICReg Loss.** The loss consists of three terms: (i) *invariance*, computed as the trace distance between within-view and cross-view Gram matrices; (ii) *variance*, computed from eigenvalues of the double-centered kernel matrix, penalizing directions with variance below a threshold; and (iii) *covariance*, computed as the squared Hilbert–Schmidt norm of the covariance operator, equivalent to the sum of squared off-diagonal entries in the kernel covariance. The implementation returns all three contributions separately, allowing monitoring of invariance, variance, and covariance during training.

**Training Setup.** We conducted experiments on MNIST, CIFAR-10, STL-10, TinyImageNet, and ImageNet100. Unless otherwise noted, the encoder was a ResNet-18 backbone (with adjusted stem for CIFAR) followed by a 3-layer MLP projector with hidden dimension 1024. Models were trained using the Adam optimizer with a learning rate of  $10^{-3}$ , a batch size of 512, and a cosine learning rate schedule. Augmentations followed standard SSL protocols: random crop, color jitter, blur, and horizontal flip for natural images; affine transforms for MNIST.

**Hyperparameters.** Table 4 reports the best-performing  $(\alpha, \beta, \zeta)$  found under our limited search for each dataset/kernel. We include these values to document the observed dataset dependence rather than to claim a single universally optimal setting. This dependence is expected because the kernel acts as an inductive bias that changes the geometry of the objective in RKHS.

**Table 4.** Best-performing  $(\alpha, \beta, \zeta)$  coefficients for Kernel VICReg across datasets.

| Kernel    | MNIST    |         |         | CIFAR-10 |         |         | STL-10   |         |         | TinyImageNet |         |         |
|-----------|----------|---------|---------|----------|---------|---------|----------|---------|---------|--------------|---------|---------|
|           | $\alpha$ | $\beta$ | $\zeta$ | $\alpha$ | $\beta$ | $\zeta$ | $\alpha$ | $\beta$ | $\zeta$ | $\alpha$     | $\beta$ | $\zeta$ |
| Linear    | 0.5      | 2       | 3       | 0.5      | 1       | 2       | 0.5      | 1       | 2       | 0.1          | 2       | 0.1     |
| RBF       | 0.5      | 2       | 2.5     | 0.5      | 1       | 3       | 0.5      | 1       | 2       | 0.1          | 2       | 0.2     |
| Laplacian | 0.5      | 2       | 3       | 0.2      | 1       | 2       | 0.5      | 1       | 2       | 0.1          | 2       | 0.1     |
| RQ        | 0.5      | 2       | 3       | 1.9      | 1       | 8       | 1.6      | 1       | 7       | 0.2          | 2       | 0.3     |

**Practical kernel-selection heuristics and reduced tuning.** As a concise rule of thumb, Laplacian typically favors sharper/local structure (edge- and texture-dominated data), RBF favors smoother geometry and can be more forgiving under higher noise, and RQ is a practical middle ground when both local and global (multi-scale) structure matter. To reduce tuning burden in practice, one can fix  $\alpha$  to a standard VICReg-scale value and perform small one-dimensional sweeps over  $\beta$  (to prevent spectral collapse relative to  $\gamma$ ) and  $\zeta$  (to discourage redundancy), rather than jointly grid-searching all  $(\alpha, \beta, \text{ and } \zeta)$ .

**Evaluation.** Representation quality was measured via *linear probing*: embeddings from the frozen encoder were extracted, and a linear classifier was trained with SGD for 100 epochs. In addition, we visualized embedding geometry using UMAP and tracked top eigenvalue dynamics of the centered kernel matrix across epochs to study variance C.

#### 5.4. Computational Complexity and Empirical Overhead

While Kernel VICReg introduces nonlinear mapping via the RKHS, it is important to quantify the overhead relative to the standard Euclidean VICReg. Let  $b$  denote the batch size and  $p$  the dimensionality of the embeddings. Standard VICReg computes a covariance matrix in  $O(bp^2 + p^3)$ , whereas Kernel VICReg computes the Gram matrix and its eigenvalue decomposition in  $O(b^2p + b^3)$ . Figures 4 and 5 further quantify how kernel choice scales in latency and memory as embedding dimension and batch size increase.

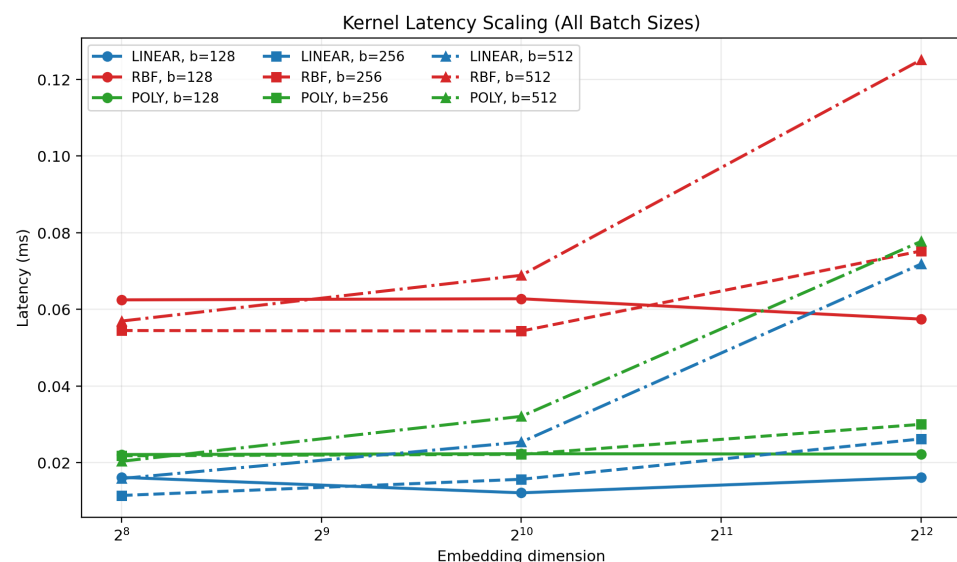


**Figure 4.** Kernel latency scaling across embedding dimensions. Each curve corresponds to a kernel type (Linear, RBF, Polynomial) at a fixed batch size, and line style differentiates batch size while color differentiates kernel family. Latency increases with both embedding dimension and batch size, with RBF consistently incurring the highest compute cost and linear kernel remaining the fastest in most regimes. This figure quantifies the computational overhead of kernel choice under high-dimensional settings.

The results indicate that for standard SSL batch sizes ( $b \leq 2048$ ), the overhead is marginal. As discussed in Section 4.7, for larger “cognitive computing” scales where  $b$  increases significantly, the  $O(b^3)$  bottleneck can be effectively mitigated using Nyström approximations or Random Fourier Features, which reduce the complexity back to a linear or quasi-linear relationship with batch size.

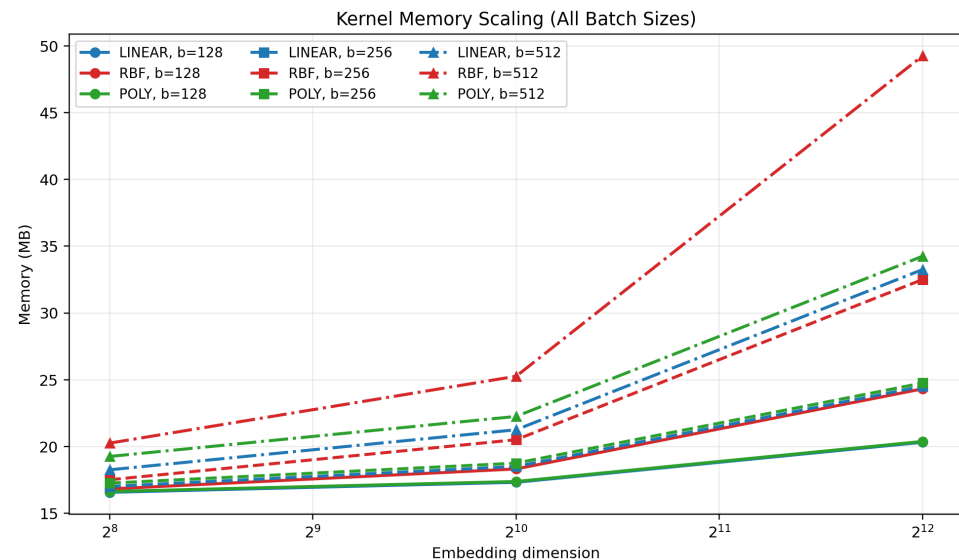


**Figure 5.** Kernel memory scaling across embedding dimensions. Color encodes kernel type (Linear, RBF, Polynomial), and line style encodes batch size. Memory usage increases with embedding dimension and batch size for all kernels, while RBF shows the highest memory footprint at larger dimensions, indicating its greater resource demand in high-dimensional regimes.

## 6. Conclusions

We introduced Kernel VICReg, a principled extension of VICReg that pulls self-supervised learning objectives from Euclidean space to Reproducing Kernel Hilbert Space (RKHS). By reformulating invariance, variance, and covariance terms in kernel space using double-centered kernel matrices and Hilbert–Schmidt norms, our method captures complex nonlinear structures without requiring explicit feature mappings.

Our empirical results across diverse datasets demonstrate the robustness and effectiveness of this kernelized formulation. Kernel VICReg outperforms its Euclidean counterpart, particularly in challenging regimes such as TinyImageNet, where standard VICReg collapses. Moreover, kernel-induced representations exhibit superior generalization in transfer learning tasks, as evidenced by improvements on STL-10. Visualization through UMAP further reveals that kernelization promotes more compact, isometric cluster structures, especially under the Laplacian kernel.

These findings suggest that kernel methods offer a natural and powerful means to enhance self-supervised learning while our work focuses on VICReg, the framework readily extends to other SSL objectives such as Barlow Twins and SimCLR, opening promising avenues for future research in kernelized SSL. This study contributes to bridging classical kernel theory and modern representation learning by showing that the integration of RKHS structure meaningfully improves both stability and expressiveness in self-supervised models.

**Author Contributions:** Conceptualization, M.H.S., B.G. and P.F.; methodology, M.H.S. and B.G.; software, M.H.S.; validation, M.H.S. and B.G.; formal analysis, M.H.S. and B.G.; investigation, M.H.S. and B.G.; writing—original draft preparation, M.H.S. and B.G.; writing—review and editing, M.H.S.,

B.G., S.M. and P.F.; visualization, M.H.S.; supervision, P.F.; project administration, S.M.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grants Program.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original data presented in the study are openly available in publicly accessible repositories: MNIST (<http://yann.lecun.com/exdb/mnist/> (accessed on 1 August 2025)), CIFAR-10 (<https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 1 August 2025)), STL-10 (<https://cs.stanford.edu/~acoates/stl10/> (accessed on 1 August 2025)), TinyImageNet (<https://www.kaggle.com/c/tiny-imagenet> (accessed on 1 August 2025)), and ImageNet (<https://www.image-net.org/> (accessed on 1 August 2025)). ImageNet100 is a subset of ImageNet constructed for evaluation in this study; no new data were created.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A. Proof for Theorem 1

### Step 1: Injectivity of Universal Kernels.

A kernel  $k$  on a compact domain is universal if its RKHS is dense in  $C(\mathcal{M})$  with respect to the supremum norm. For universal kernels such as the Gaussian RBF and Laplacian kernels, the associated feature map  $\phi$  is injective; that is:

$$x \neq y \implies \phi(x) \neq \phi(y).$$

Hence, the embedding  $\phi : \mathcal{M} \rightarrow \mathcal{H}$  is one-to-one and preserves all information about the manifold.

### Step 2: Linearization of Nonlinear Structure.

Although  $\mathcal{M}$  may be nonlinear in  $\mathbb{R}^p$ , its image  $\phi(\mathcal{M})$  lies in a (possibly infinite-dimensional) Hilbert space  $\mathcal{H}$ , where linear combinations are permitted. Kernel PCA theory [15] shows that principal components in RKHS correspond to eigenfunctions of the covariance operator:

$$C_\phi(x) = \frac{1}{b} \sum_{i=1}^b (\phi(x_i) - \bar{\phi})(\phi(x_i) - \bar{\phi})^\top.$$

The eigenvalue problem in  $\mathcal{H}$  reduces to the finite-dimensional eigenvalue problem of the centered kernel matrix:

$$\hat{K}(x)v_i = \lambda_i v_i.$$

Thus, eigenvalues  $\{\lambda_i\}$  represent variances along nonlinear principal directions in feature space.

### Step 3: Failure of Euclidean Covariance.

In Euclidean space, covariance captures only second-order linear variation:

$$C_{\text{Euc}} = \frac{1}{b} Z^\top H Z.$$

If  $\mathcal{M}$  exhibits curvature or nonlinear embedding (e.g., Swiss roll or concentric circles), Euclidean PCA cannot flatten the manifold; variance is spread across multiple coordinates without revealing intrinsic nonlinear directions. Thus, Euclidean variance regularization may fail to preserve intrinsic manifold modes.

**Step 4: RKHS Variance Captures Nonlinear Modes.**

In contrast, kernel PCA diagonalizes the covariance operator in  $\mathcal{H}$ . Since universal kernels generate feature spaces dense in  $C(\mathcal{M})$ , nonlinear structure becomes linearly separable in  $\mathcal{H}$ . Therefore, eigenvalues  $\{\lambda_i\}$  of  $\hat{K}(x)$  quantify nonlinear principal variances of  $\mathcal{M}$ . Enforcing  $\lambda_i/b \geq \gamma^2 > 0$  ensures preservation of nonlinear modes of variation.

**Conclusion.**

Because RKHS covariance eigenvalues correspond to nonlinear principal components, kernelized variance regularization preserves nonlinear structure that Euclidean covariance cannot capture.

**Appendix B. Proof for Theorem 2****Step 1: Decomposition of the Gram Matrix.**

Define feature map  $\phi(x)$  in RKHS  $\mathcal{H}$ . The (uncentered) Gram matrix satisfies:

$$K_b[i, j] = \langle \phi(x_i), \phi(x_j) \rangle.$$

Let  $\Phi = [\phi(x_1), \dots, \phi(x_b)]$ . Then, we have  $K_b = \Phi^\top \Phi$ . The centered Gram matrix is:

$$\hat{K}_b = H\Phi^\top \Phi H = (\Phi H)^\top (\Phi H).$$

Thus,  $\hat{K}_b$  is the finite-sample covariance operator expressed in kernel coordinates.

**Step 2: Operator Representation.**

Define centered feature vectors:

$$\tilde{\phi}(x_i) = \phi(x_i) - \mu, \quad \mu = \mathbb{E}[\phi(x)].$$

The population covariance operator in RKHS is  $C(x) := \mathbb{E}[\tilde{\phi}(x) \otimes \tilde{\phi}(x)]$ . The empirical covariance operator  $C_b(x)$  is  $(1/b) \sum_{i=1}^b \tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i)$ , as also stated in Equation (10). Moreover, according to the definition in the theorem, we have  $\Sigma := \mathbb{E}[\hat{K}_b]$ . Thus:

$$\begin{aligned} \hat{K}_b &= bC_b, \\ \Sigma &= bC. \end{aligned}$$

Thus, the bounding  $\|\hat{K}_b - \Sigma\|_2$  reduces to bounding operator deviation  $\|C_b - C\|_2$ .

**Step 3: Boundedness.**

According to Equation (34), the kernel is bounded:

$$\|\phi(x)\|_{\mathcal{H}}^2 = k(x, x) \leq \kappa^2,$$

therefore,

$$\|\tilde{\phi}(x)\|_{\mathcal{H}} \leq 2\kappa.$$

Hence, each rank-one operator,

$$X_i := \tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i),$$

satisfies

$$\|X_i\|_2 = \|\tilde{\phi}(x_i)\|^2 \leq 4\kappa^2.$$

**Step 4: Apply Matrix Bernstein Inequality.**

The operators:

$$Y_i = X_i - \mathbb{E}[X_i],$$

are independent mean-zero self-adjoint operators. Matrix Bernstein inequality [24] states that for such operators:

$$\Pr\left(\left\|\frac{1}{b} \sum_{i=1}^b Y_i\right\|_2 \geq t\right) \leq 2b \exp\left(\frac{-bt^2/2}{\sigma^2 + Rt/3}\right),$$

where

$$R = \max_i \|Y_i\|_2 \leq 8\kappa^2,$$

and variance proxy

$$\sigma^2 \leq \frac{1}{b} \sum_{i=1}^b \mathbb{E}[\|Y_i\|_2^2] \leq 4\kappa^4.$$

Solving the inequality for probability  $\delta$  gives:

$$\|C_b - C\|_2 \leq c\kappa^2 \left( \sqrt{\frac{\log(2b/\delta)}{b}} + \frac{\log(2b/\delta)}{b} \right).$$

#### Step 5: Transfer to Kernel Matrix.

According to step 2 of the proof, the bounding  $\|\hat{K}_b - \Sigma\|_2$  reduces to bounding operator deviation  $\|C_b - C\|_2$ . Therefore, we obtain the same deviation bound for  $\|\hat{K}_b - \Sigma\|_2$ .

## References

1. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 1597–1607.
2. Bardes, A.; Ponce, J.; LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *Proceedings of the International Conference on Learning Representations*, Virtual, 25–29 April 2022.
3. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent—a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
4. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow Twins: Self-supervised learning via redundancy reduction. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 12310–12320.
5. Li, Y.; Pogodin, R.; Sutherland, D.J.; Gretton, A. Self-supervised learning with kernel dependence maximization. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15543–15556.
6. Wu, Y.; Greenspan, M. Pseudo-Keypoint RKHS Learning for Self-supervised 6DoF Pose Estimation. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 37–56.
7. Ni, X.; Xiong, F.; Zheng, Y.; Wang, L. Graph Contrastive Learning with Kernel Dependence Maximization for Social Recommendation. In *Proceedings of the ACM on Web Conference 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 481–492.
8. Sepanj, M.H.; Ghojogh, B.; Fieguth, P. Self-supervised learning using nonlinear dependence. *IEEE Access* **2025**, *13*, 190582–190589. [CrossRef]
9. Sepanj, H.; Fieguth, P. Aligning Feature Distributions in VICReg Using Maximum Mean Discrepancy for Enhanced Manifold Awareness in Self-Supervised Representation Learning. *J. Comput. Vis. Imaging Syst.* **2024**, *10*, 13–18.
10. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
11. Gretton, A.; Bousquet, O.; Smola, A.; Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings of the International Conference on Algorithmic Learning Theory*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 63–77.
12. Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; Smola, A. A kernel method for the two-sample-problem. *Adv. Neural Inf. Process. Syst.* **2006**, *19*, 513–520.

13. Simon, J.B.; Knutins, M.; Ziyin, L.; Geisz, D.; Fetterman, A.J.; Albrecht, J. On the stepwise nature of self-supervised learning. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2023; pp. 31852–31876.
14. Schölkopf, B.; Smola, A.; Müller, K.R. Kernel principal component analysis. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 583–588.
15. Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319. [[CrossRef](#)]
16. Schölkopf, B. The kernel trick for distances. *Adv. Neural Inf. Process. Syst.* **2000**, *13*.
17. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K.R. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No. 98th8468)*; IEEE: Piscataway, NJ, USA, 1999; pp. 41–48.
18. Berahmand, K.; Saberi-Movahed, F.; Sheikhpour, R.; Li, Y.; Jalili, M. A comprehensive survey on spectral clustering with graph structure learning. *arXiv* **2025**, arXiv:2501.13597. [[CrossRef](#)]
19. Zhu, Y.; Xu, Y.; Liu, Q.; Wu, S. An empirical study of graph contrastive learning. *arXiv* **2021**, arXiv:2109.01116. [[CrossRef](#)]
20. Nyström, E.J. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Math.* **1930**, *54*, 185–204. [[CrossRef](#)]
21. Williams, C.; Seeger, M. Using the Nyström method to speed up kernel machines. *Adv. Neural Inf. Process. Syst.* **2000**, *13*.
22. Rahimi, A.; Recht, B. Random Features for Large-Scale Kernel Machines. *Adv. Neural Inf. Process. Syst.* **2007**, *20*.
23. Liang, Z.; Luo, Y.; Beese, M.; Drexlin, D.J. Multiple Positive Views in Self-Supervised Learning. 2024. Available online: <https://openreview.net/forum?id=WGP2pHtLtn> (accessed on 2 March 2026)
24. Tropp, J.A. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.* **2012**, *12*, 389–434. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.